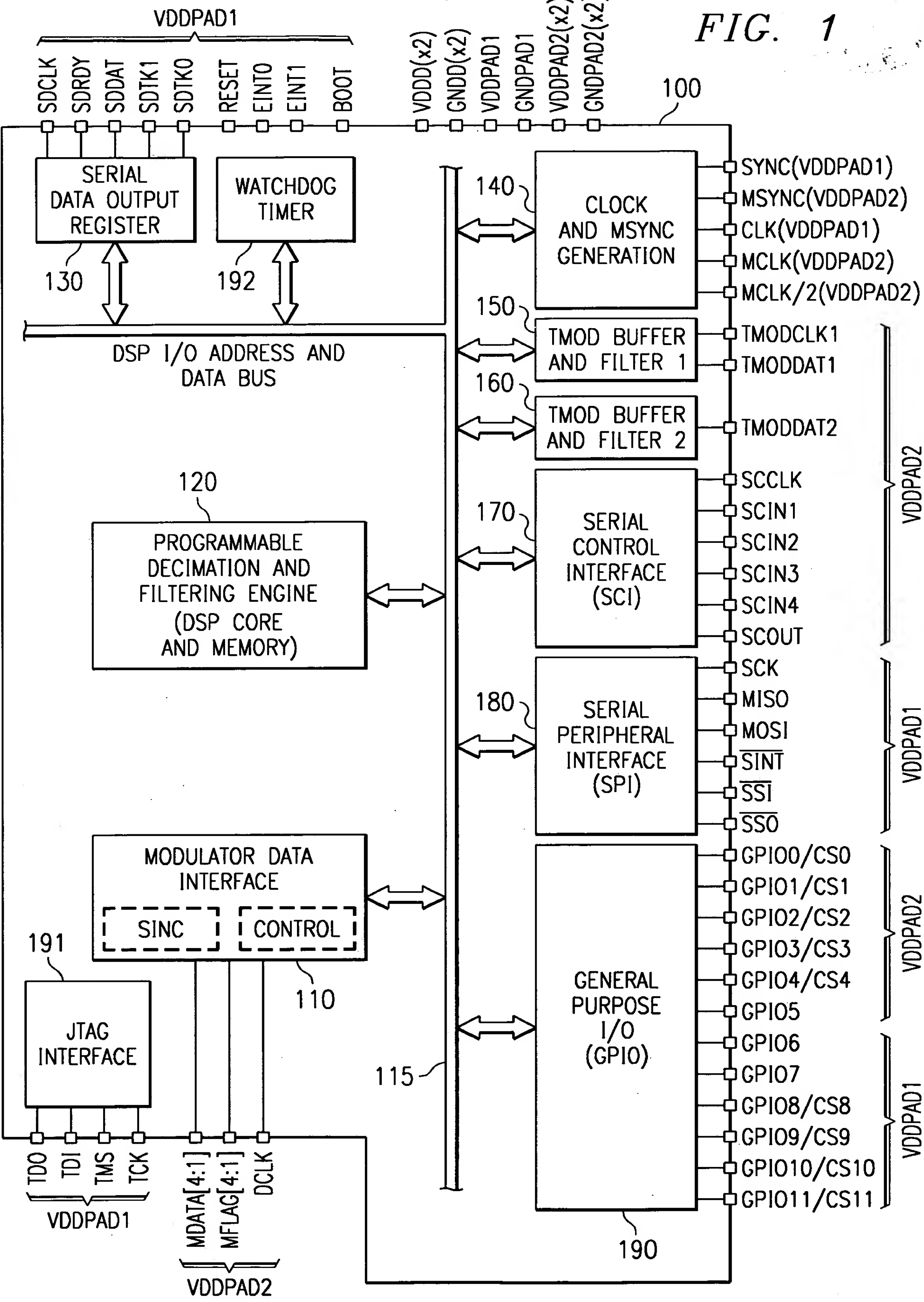
FIG. 1


VDDPAD1
SDCLK
SDRDY
SDDAT
SDTK1
SDTK0
RESET
EINT0
EINT1
BOOT
VDDD(x2)
GNDD(x2)
VDDPAD1
GNDPAD1
VDDPAD2(x2)
GNDPAD2(x2)
100
SERIAL DATA OUTPUT REGISTER
130
WATCHDOG TIMER
192
DSP I/O ADDRESS AND DATA BUS
140
CLOCK AND MSYNC GENERATION
SYNC(VDDPAD1)
MSYNC(VDDPAD2)
CLK(VDDPAD1)
MCLK(VDDPAD2)
MCLK/2(VDDPAD2)
150
TMOD BUFFER AND FILTER 1
TMODCLK1
TMODDAT1
160
TMOD BUFFER AND FILTER 2
TMODDAT2
VDDPAD2
170
SERIAL CONTROL INTERFACE (SCI)
SCCLK
SCIN1
SCIN2
SCIN3
SCIN4
SCOUT
120
PROGRAMMABLE DECIMATION AND FILTERING ENGINE (DSP CORE AND MEMORY)
180
SERIAL PERIPHERAL INTERFACE (SPI)
SCK
MISO
MOSI
$\overline{\text{SINT}}$
$\overline{\text{SSI}}$
$\overline{\text{SSO}}$
VDDPAD1
MODULATOR DATA INTERFACE
SINC
CONTROL
110
191
JTAG INTERFACE
115
GENERAL PURPOSE I/O (GPIO)
GPIO0/CS0
GPIO1/CS1
GPIO2/CS2
GPIO3/CS3
GPIO4/CS4
GPIO5
VDDPAD2
GPIO6
GPIO7
GPIO8/CS8
GPIO9/CS9
GPIO10/CS10
GPIO11/CS11
VDDPAD1
190
TDO
TDI
TMS
TCK
VDDPAD1
MDATA[4:1]
MFLAG[4:1]
DCLK
VDDPAD2

]

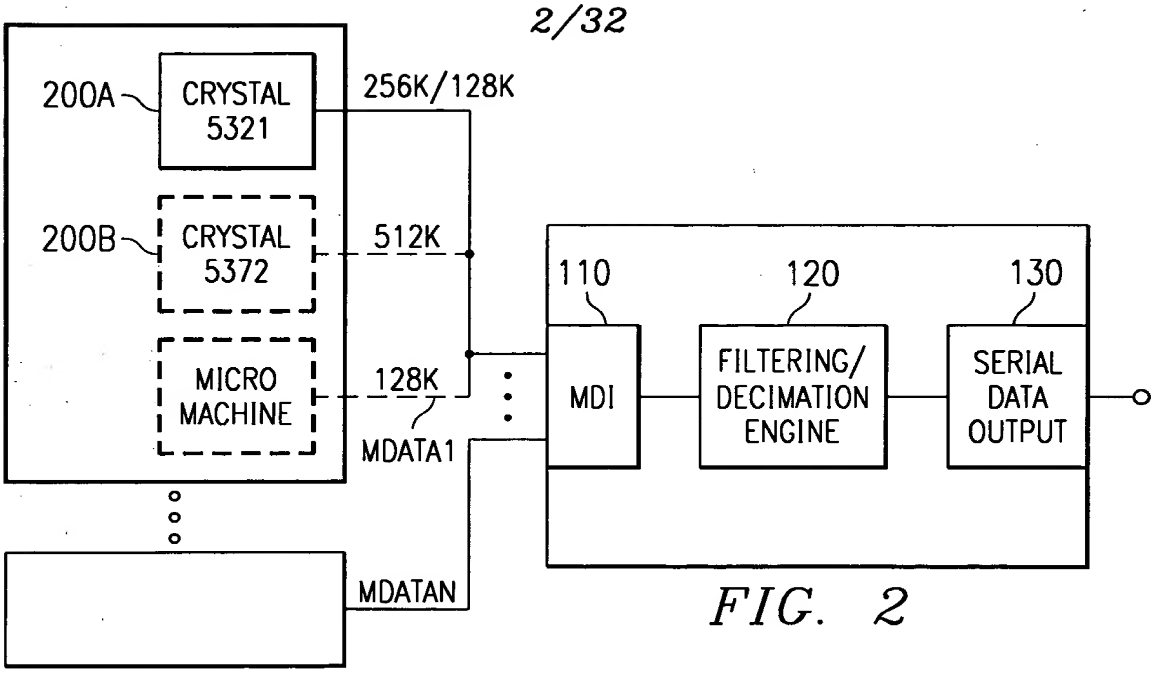


FIG. 2

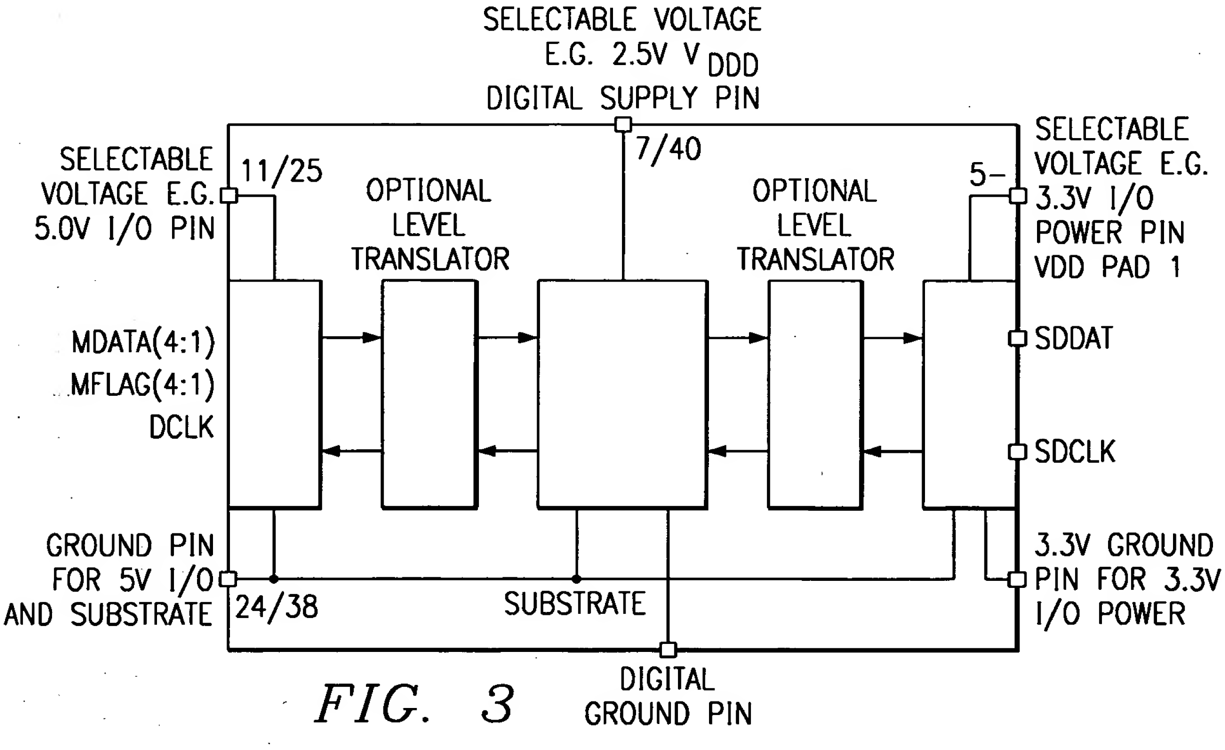

SELECTABLE VOLTAGE E.G. 2.5V V DDD DIGITAL SUPPLY PIN
7/40
SELECTABLE VOLTAGE E.G. 5.0V I/O PIN
11/25
OPTIONAL LEVEL TRANSLATOR
OPTIONAL LEVEL TRANSLATOR
5
SELECTABLE VOLTAGE E.G. 3.3V I/O POWER PIN VDD PAD 1
MDATA(4:1)
MFLAG(4:1)
DCLK
SDDAT
SDCLK
GROUND PIN FOR 5V I/O AND SUBSTRATE
24/38
SUBSTRATE
3.3V GROUND PIN FOR 3.3V I/O POWER
DIGITAL GROUND PIN

FIG. 3

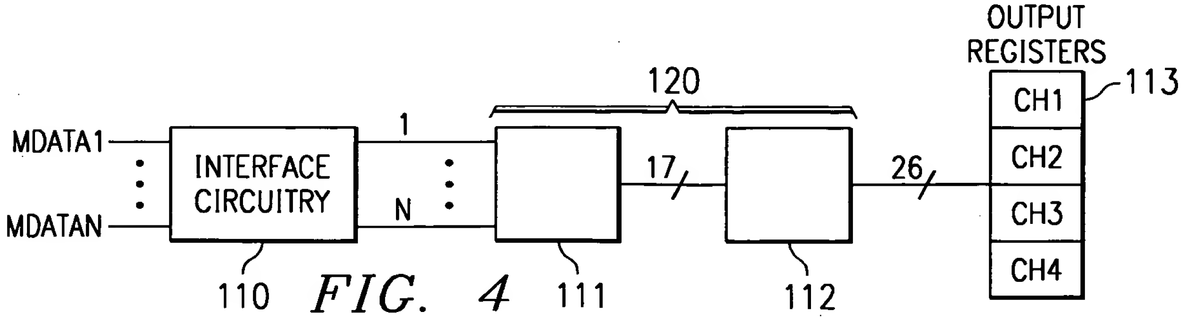


FIG. 4

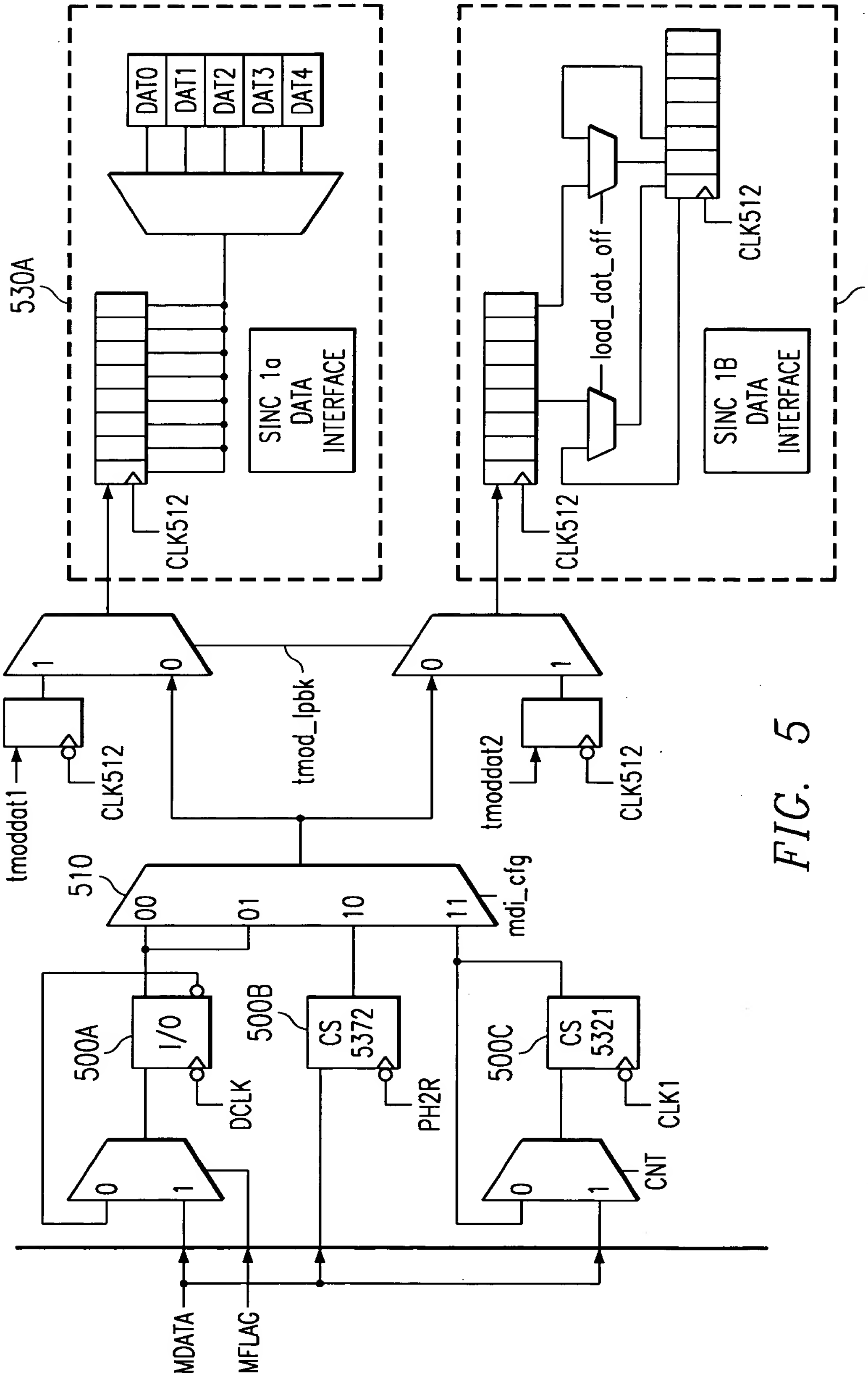

530A
DAT0
DAT1
DAT2
DAT3
DAT4
CLK512
SINC 1a DATA INTERFACE
530B
load_dat_off
CLK512
CLK512
SINC 1B DATA INTERFACE
1
0
0
1
tmod_lpbk
tmoddat1
CLK512
tmoddat2
CLK512
510
00
01
10
11
mdi_cfg
500A
I/O
DCLK
500B
CS 5372
PH2R
500C
CS 5321
CLK1
0
1
0
1
CNT
MDATA
MFLAG

FIG. 5

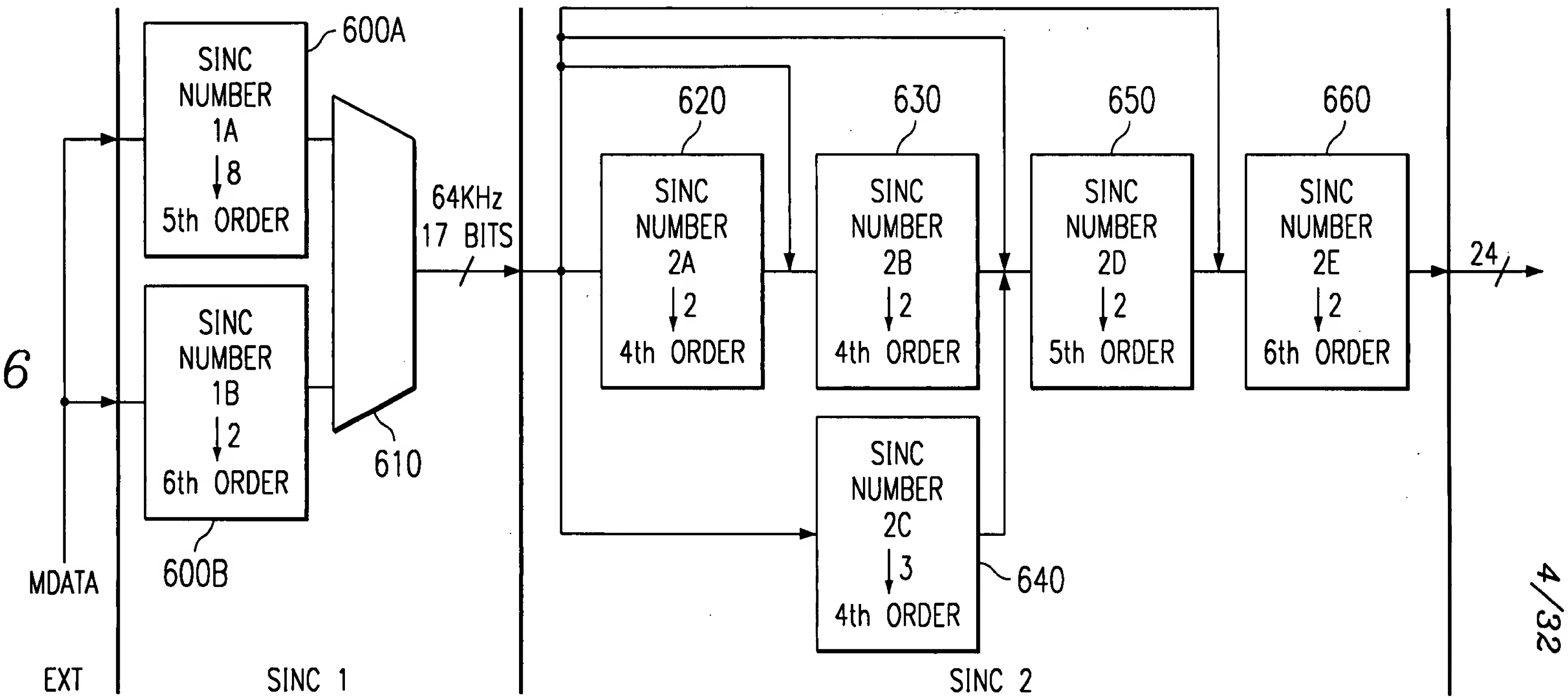

600A
SINC NUMBER 1A ↓8 5th ORDER
SINC NUMBER 1B ↓2 6th ORDER
600B
610
64KHz 17 BITS
MDATA
EXT
SINC 1
620
SINC NUMBER 2A ↓2 4th ORDER
630
SINC NUMBER 2B ↓2 4th ORDER
SINC NUMBER 2C ↓3 4th ORDER
640
650
SINC NUMBER 2D ↓2 5th ORDER
660
SINC NUMBER 2E ↓2 6th ORDER
24
SINC 2

FIG. 6

FIG. 7

- FIFTH ORDER DECIMATE BY 8:

$$H(z)=\left(\frac{1-z^{-8}}{1-z^{-1}}\right)^{5}$$

- 36 TAP FIR FILTER. HALF OF THE (SYMMETRIC) COEFFICIENTS

$h_0=1$	$h_1=5$	$h_2=15$	$h_3=35$	$h_4=70$	$h_5=126$	$h_6=210$	$h_7=330$	$h_8=490$
$h_9=690$	$h_{10}=926$	$h_{11}=1190$	$h_{12}=1470$	$h_{13}=1750$	$h_{14}=2010$	$h_{15}=2226$	$h_{16}=2380$	$h_{17}=2460$

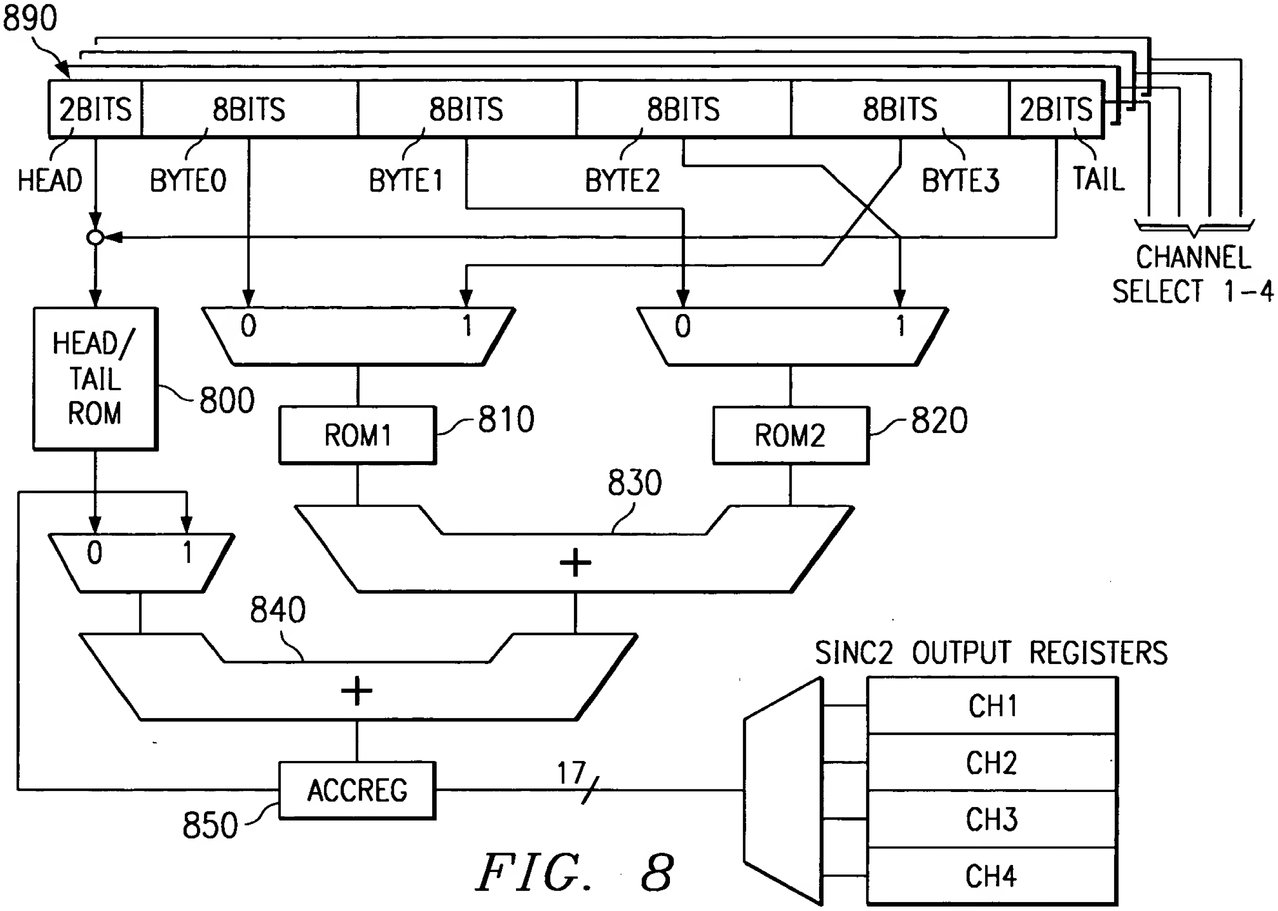


FIG. 8

$$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^{6}$$

IMPULSE RESPONSE:

$$y[n]=x[n]+6\cdot x[n-1]+15\cdot x[n-2]+20\cdot x[n-3]+15\cdot x[n-4]+6\cdot x[n-5]+x[n-6]$$

FIG. 9

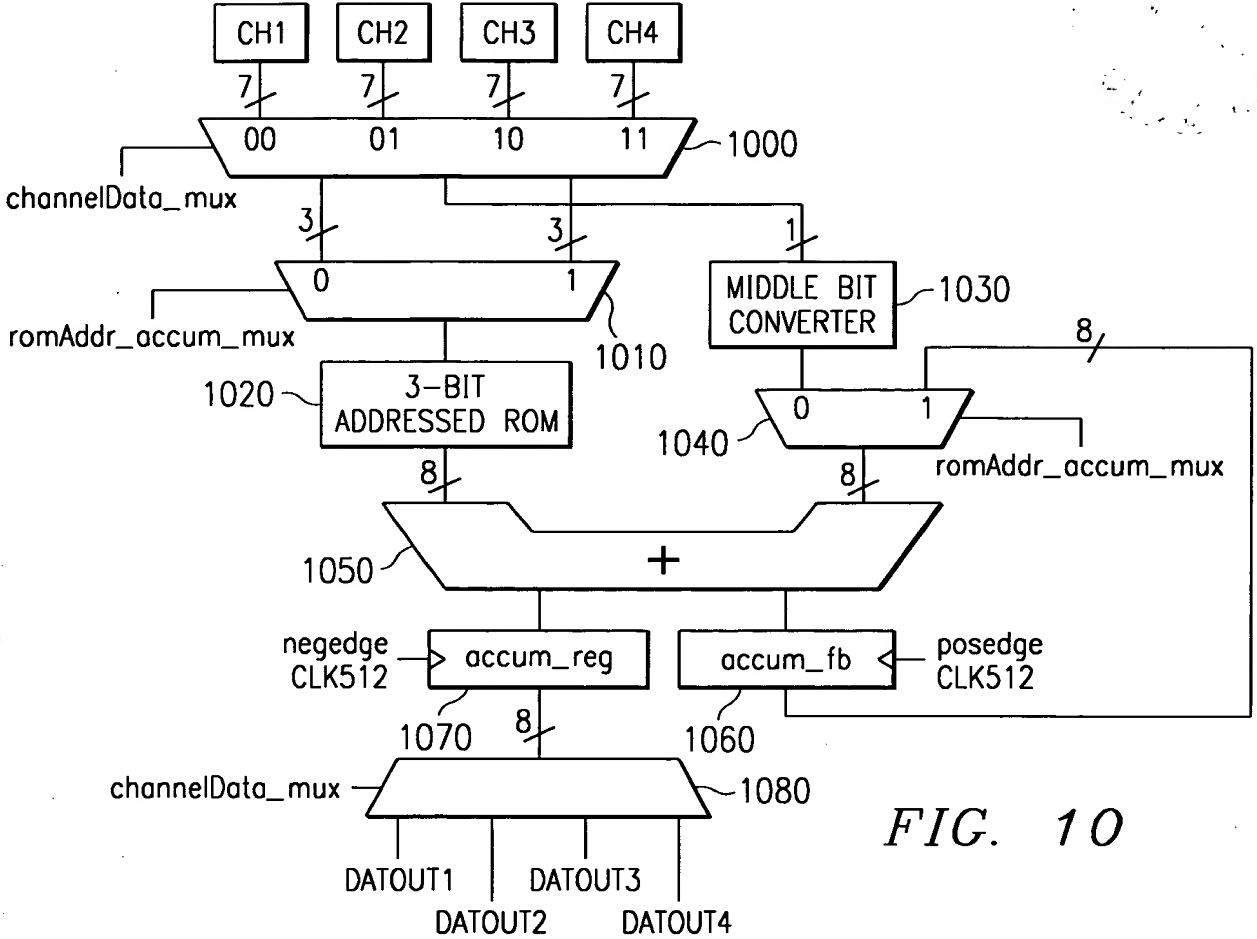


FIG. 10

FILTER NAME	SYSTEM FUNCTION	IMPULSE RESPONSE (FILTER COEFFICIENTS)

Sinc2(a)
Sinc2(b)
	$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^4$	h[n]=[1 4 6 4 1]

Sinc2(c)	$H(z)=\left(\frac{1-z^{-3}}{1-z^{-1}}\right)^4$	h[n]=[1 4 10 16 19 16 10 4 1]
Sinc2(d)	$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^5$	h[n]=[1 5 10 10 5 1]
Sinc2(e)	$H(z)=\left(\frac{1-z^{-2}}{1-z^{-1}}\right)^6$	h[n]=[1 6 15 20 15 6 1]

FIG. 11

Sinc2(a) and Sinc2(b):

$$y[n]=x[n]+4x[n-1]+6x[n-2] \quad +4x[n-3]+x[n-4]$$
$$=x[n]+4x[n-1]+4x[n-2]+2x[n-2] \quad +4x[n-3]+x[n-4]$$

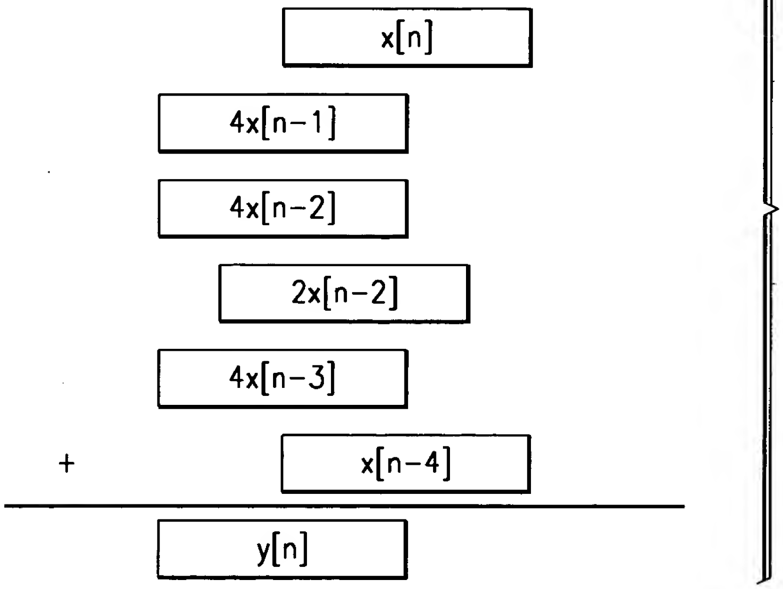


FIG. 12

Sinc2(a) and Sinc2(b):

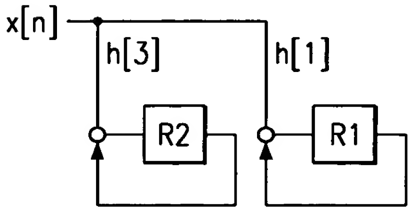


FIG. 14A

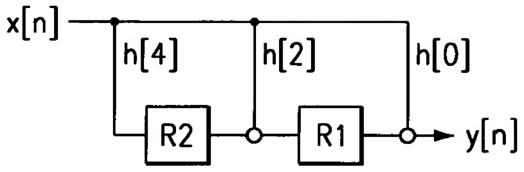


FIG. 14B

Sinc2(d):

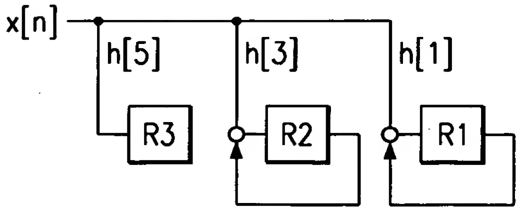


FIG. 15A

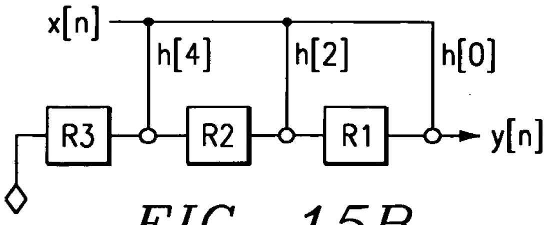


FIG. 15B

FIG. 13A

Sinc2(c):

$$
\begin{aligned}
y[n] &= x[n]+4x[n-1]+10x[n-2]+16x[n-3]+19x[n-4]+16x[n-5]+10x[n-6]+4x[n-7]+x[n-8] \\
&= x[n]+4x[n-1]+\boxed{8x[n-2]+2x[n-2]}+16x[n-3]+\boxed{16x[n-4]+2x[n-4]+x[n-4]} \\
&\quad +16x[n-5]+\boxed{8x[n-6]+2x[n-6]}+4x[n-7]+x[n-8]
\end{aligned}
$$

FIG. 13B

Sinc2(d):

$$
\begin{aligned}
y[n] &= x[n]+5x[n-1]+10x[n-2]+10x[n-3]+5x[n-4]+x[n-5] \\
&= x[n]+\boxed{4x[n-1]+x[n-1]}+\boxed{8x[n-2]+2x[n-2]}+\boxed{8x[n-3]+2x[n-3]}+\boxed{4x[n-4]+x[n-4]}+x[n-5]
\end{aligned}
$$

FIG. 13C

Sinc2(e):

$$
\begin{aligned}
y[n] &= x[n]+6x[n-1]+15x[n-2]+20x[n-3]+15x[n-4]+6x[n-5]+x[n-6] \\
&= x[n]+\boxed{4x[n-1]+2x[n-1]}+\boxed{16x[n-2]-x[n-2]}+\boxed{16x[n-3]+4x[n-3]} \\
&\quad +\boxed{16x[n-4]-x[n-4]}+\boxed{4x[n-5]+2x[n-5]}+x[n-6]
\end{aligned}
$$

Sinc2(c):

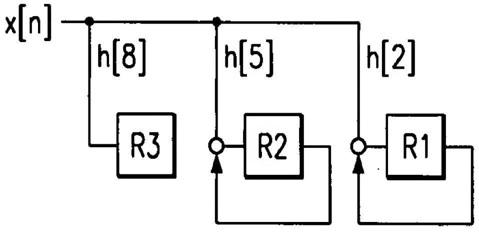


FIG. 16A

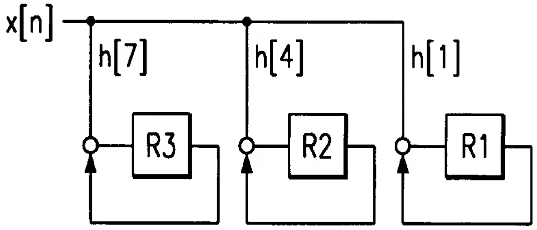


FIG. 16B

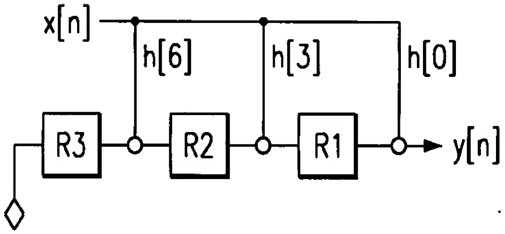


FIG. 16C

Sinc2(e):

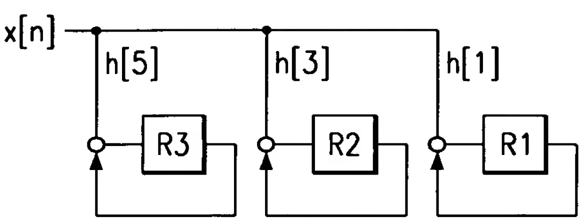


FIG. 17A

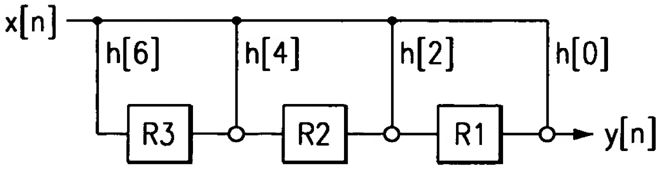


FIG. 17B

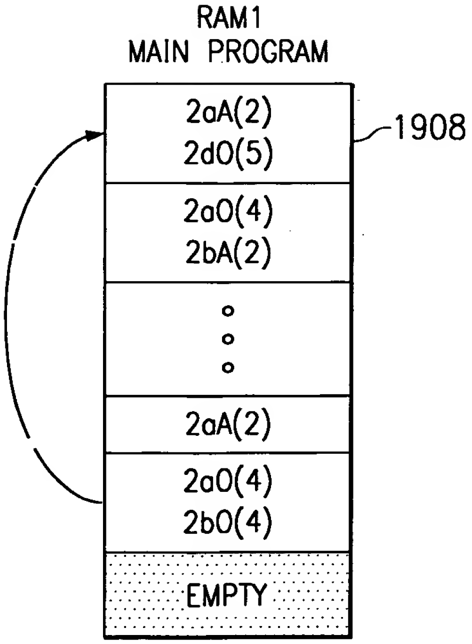


FIG. 19A

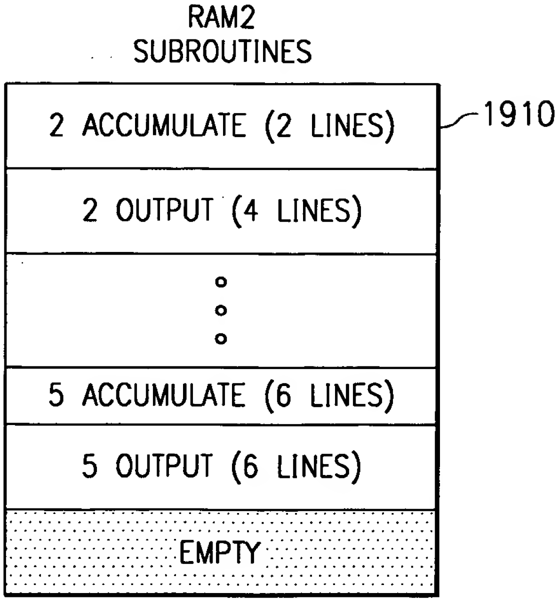


FIG. 19B

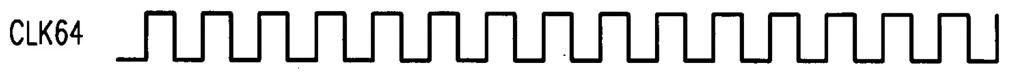


FIG. 18A

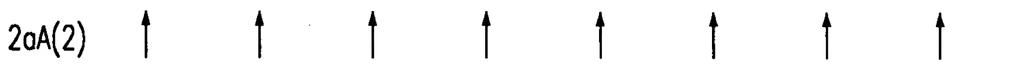


FIG. 18B

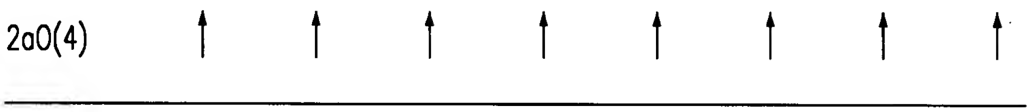


FIG. 18C

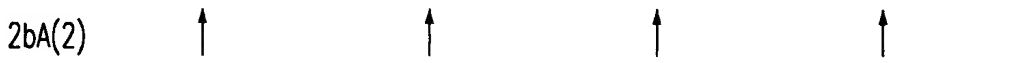


FIG. 18D

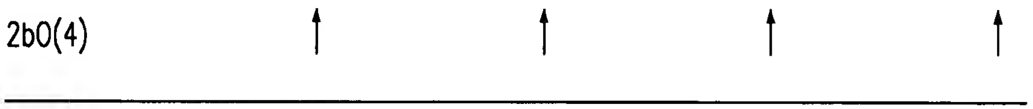


FIG. 18E

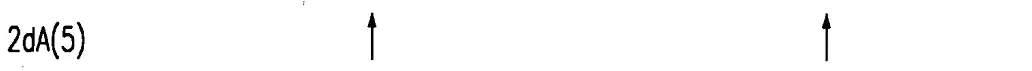


FIG. 18F

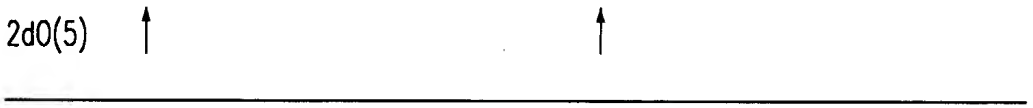


FIG. 18G

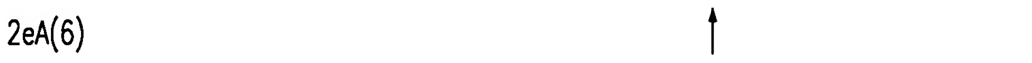


FIG. 18H

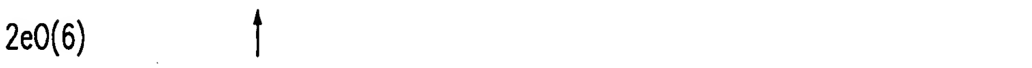


FIG. 18I

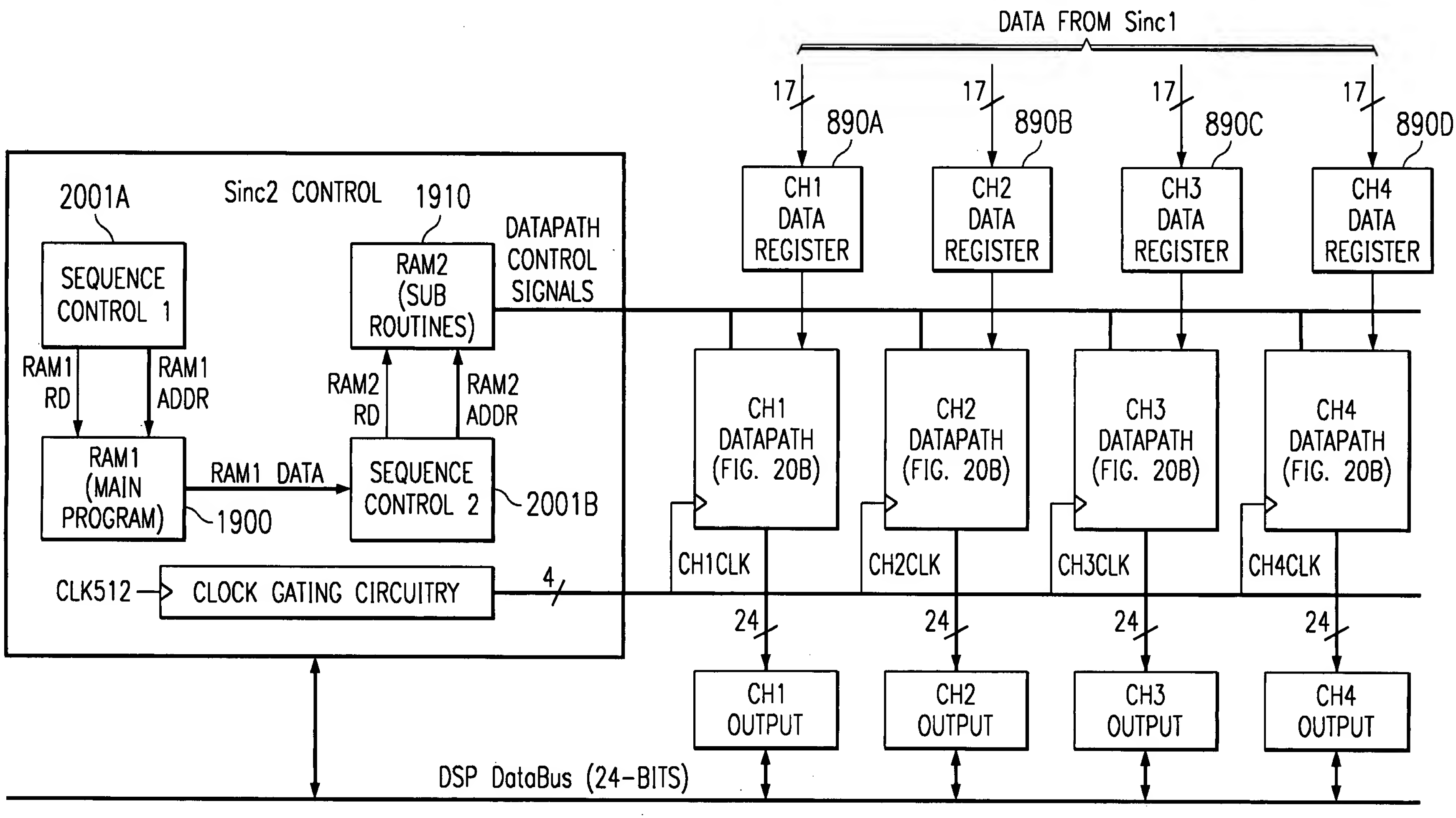

DATA FROM Sinc1
17
890A
17
890B
17
890C
17
890D
CH1 DATA REGISTER
CH2 DATA REGISTER
CH3 DATA REGISTER
CH4 DATA REGISTER
2001A
Sinc2 CONTROL
1910
DATAPATH CONTROL SIGNALS
SEQUENCE CONTROL 1
RAM2 (SUB ROUTINES)
RAM1 RD
RAM1 ADDR
RAM2 RD
RAM2 ADDR
RAM1 (MAIN PROGRAM)
RAM1 DATA
SEQUENCE CONTROL 2
1900
2001B
CH1 DATAPATH (FIG. 20B)
CH2 DATAPATH (FIG. 20B)
CH3 DATAPATH (FIG. 20B)
CH4 DATAPATH (FIG. 20B)
CH1CLK
CH2CLK
CH3CLK
CH4CLK
CLK512
CLOCK GATING CIRCUITRY
4
24
24
24
24
CH1 OUTPUT
CH2 OUTPUT
CH3 OUTPUT
CH4 OUTPUT
DSP DataBus (24-BITS)

FIG. 20A

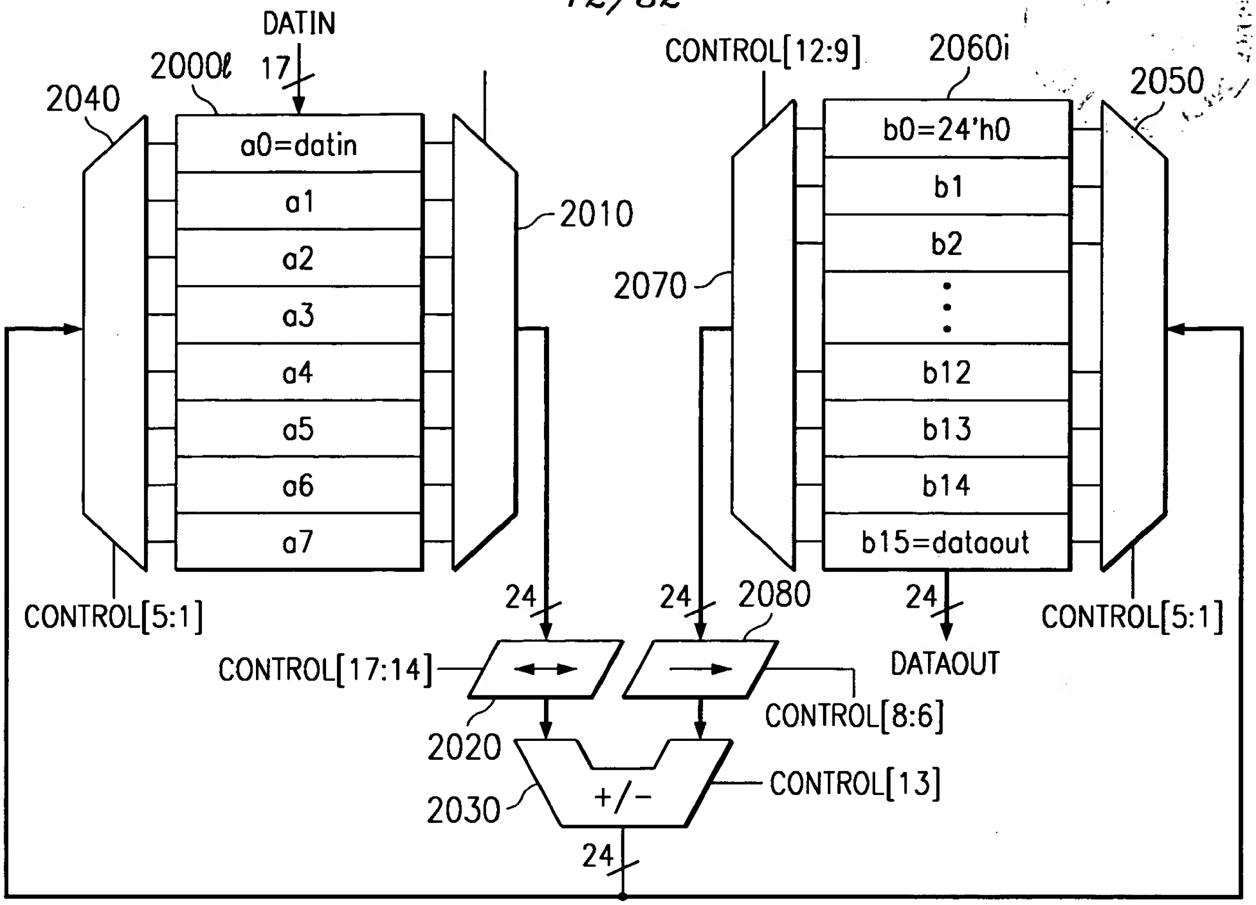


FIG. 20B

PROGRAMMING PROCEDURE:

1. SELECT DECIMATION RATE.
2. SELECT REQUIRED MINI-SINCS AND ASSOCIATAED ACCUMULATE AND OUTPUT SUBROUTINES.
3. SEPARATE COEFFICIENTS INTO FORM SUITABLE FOR SHIFT-ADD OPERATIONS.
4. CHECK FOR OVERFLOW AFTER EACH ADDITION IN THE FILTER.
5. PERFORM NECESSARY TRUNCATION TO 24 BITS AND SCALING OF SUBSEQUENT COEFFICIENTS IN MINI-SINCS.
6. TIME MULTIPLEX ACCUMULATE AND OUTPUT SUBROUTINES SO THAT A MAXIMUM OF 8 ADDITIONS/SUBTRACTIONS ARE PERFORMED FOR EACH INPUT FROM SINC1.
7. CREATE CODE FOR RAM2 (ACCUMULATE AND OUTPUT SUBROUTINES) IN THE FORM: [Coeff 1] [Src 1] [Src 2] [Dest] [Coeff2] [Done Subroutine]
8. CREATE CODE FOR RAM1 (MAIN CONTROL CODE) [Line #] [Wait for new data] [Done program]

FIG. 21

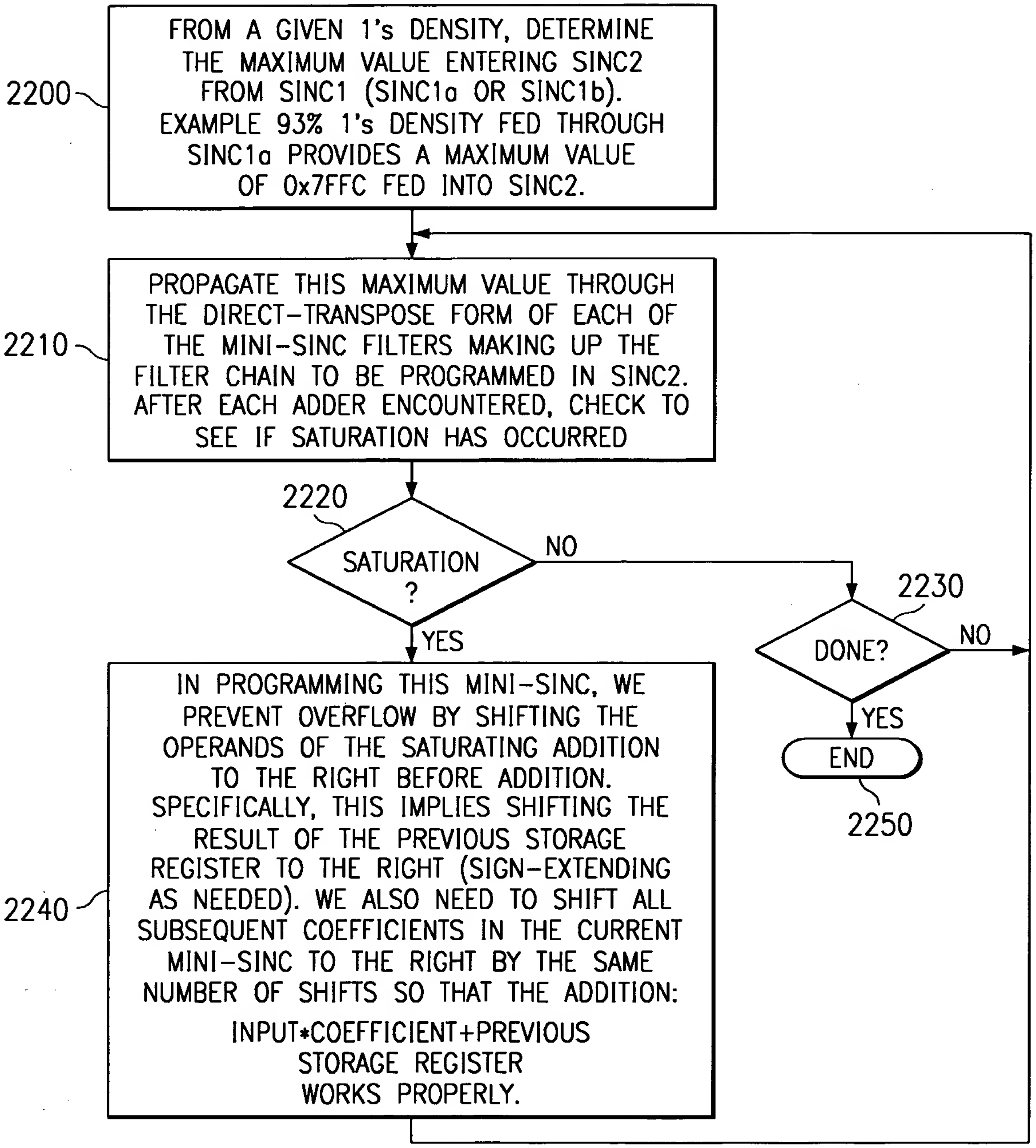

FROM A GIVEN 1's DENSITY, DETERMINE THE MAXIMUM VALUE ENTERING SINC2 FROM SINC1 (SINC1a OR SINC1b). EXAMPLE 93% 1's DENSITY FED THROUGH SINC1a PROVIDES A MAXIMUM VALUE OF 0x7FFC FED INTO SINC2.
2200
PROPAGATE THIS MAXIMUM VALUE THROUGH THE DIRECT-TRANSPOSE FORM OF EACH OF THE MINI-SINC FILTERS MAKING UP THE FILTER CHAIN TO BE PROGRAMMED IN SINC2. AFTER EACH ADDER ENCOUNTERED, CHECK TO SEE IF SATURATION HAS OCCURRED
2210
2220
SATURATION ?
NO
YES
2230
DONE?
NO
YES
END
2250
IN PROGRAMMING THIS MINI-SINC, WE PREVENT OVERFLOW BY SHIFTING THE OPERANDS OF THE SATURATING ADDITION TO THE RIGHT BEFORE ADDITION. SPECIFICALLY, THIS IMPLIES SHIFTING THE RESULT OF THE PREVIOUS STORAGE REGISTER TO THE RIGHT (SIGN-EXTENDING AS NEEDED). WE ALSO NEED TO SHIFT ALL SUBSEQUENT COEFFICIENTS IN THE CURRENT MINI-SINC TO THE RIGHT BY THE SAME NUMBER OF SHIFTS SO THAT THE ADDITION:
INPUT*COEFFICIENT+PREVIOUS STORAGE REGISTER WORKS PROPERLY.
2240

FIG. 22

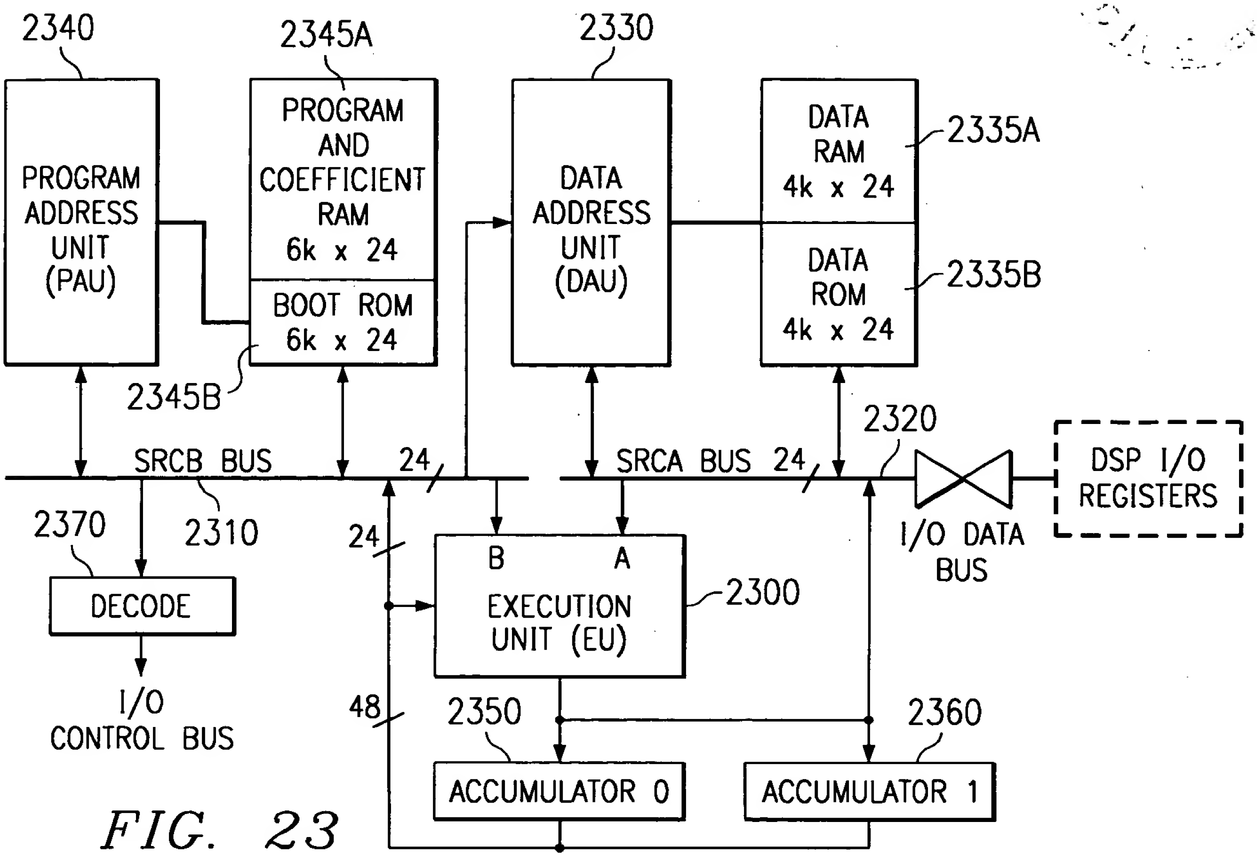


FIG. 23

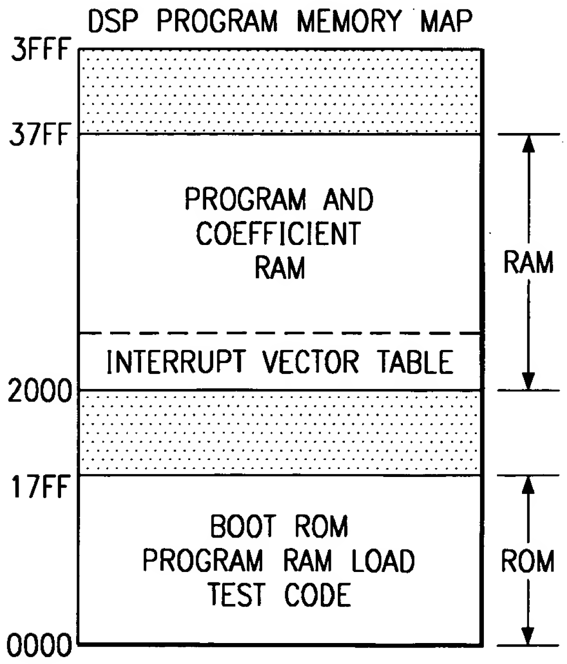


FIG. 24A

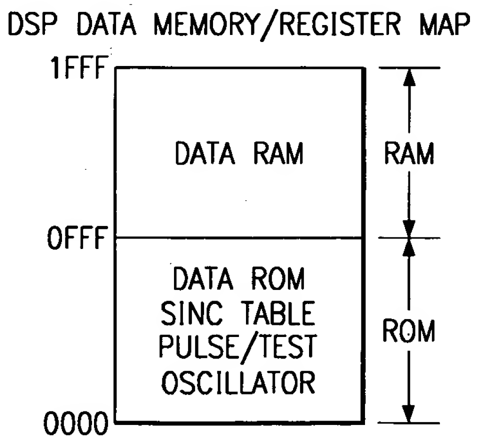

DSP DATA MEMORY/REGISTER MAP
1FFF
DATA RAM
RAM
0FFF
DATA ROM SINC TABLE PULSE/TEST OSCILLATOR
ROM
0000

FIG. 24B

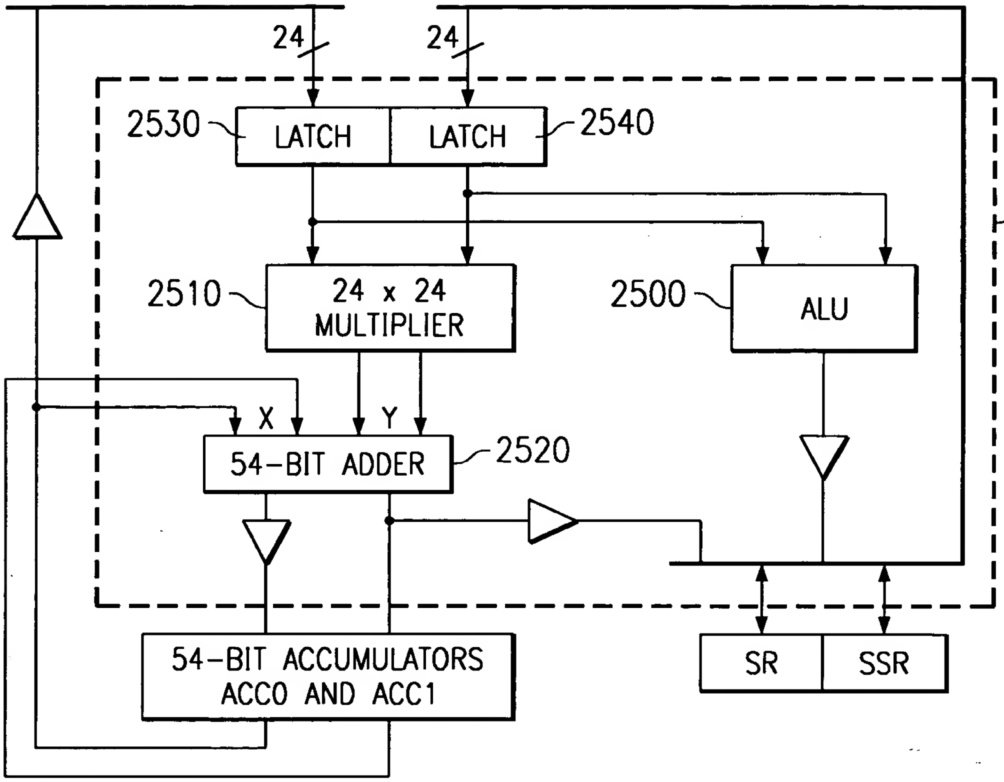

SRCB
SRCA
24
24
2530
LATCH
LATCH
2540
2300
2510
24 x 24 MULTIPLIER
2500
ALU
X
Y
54-BIT ADDER
2520
54-BIT ACCUMULATORS ACC0 AND ACC1
SR
SSR

FIG. 25

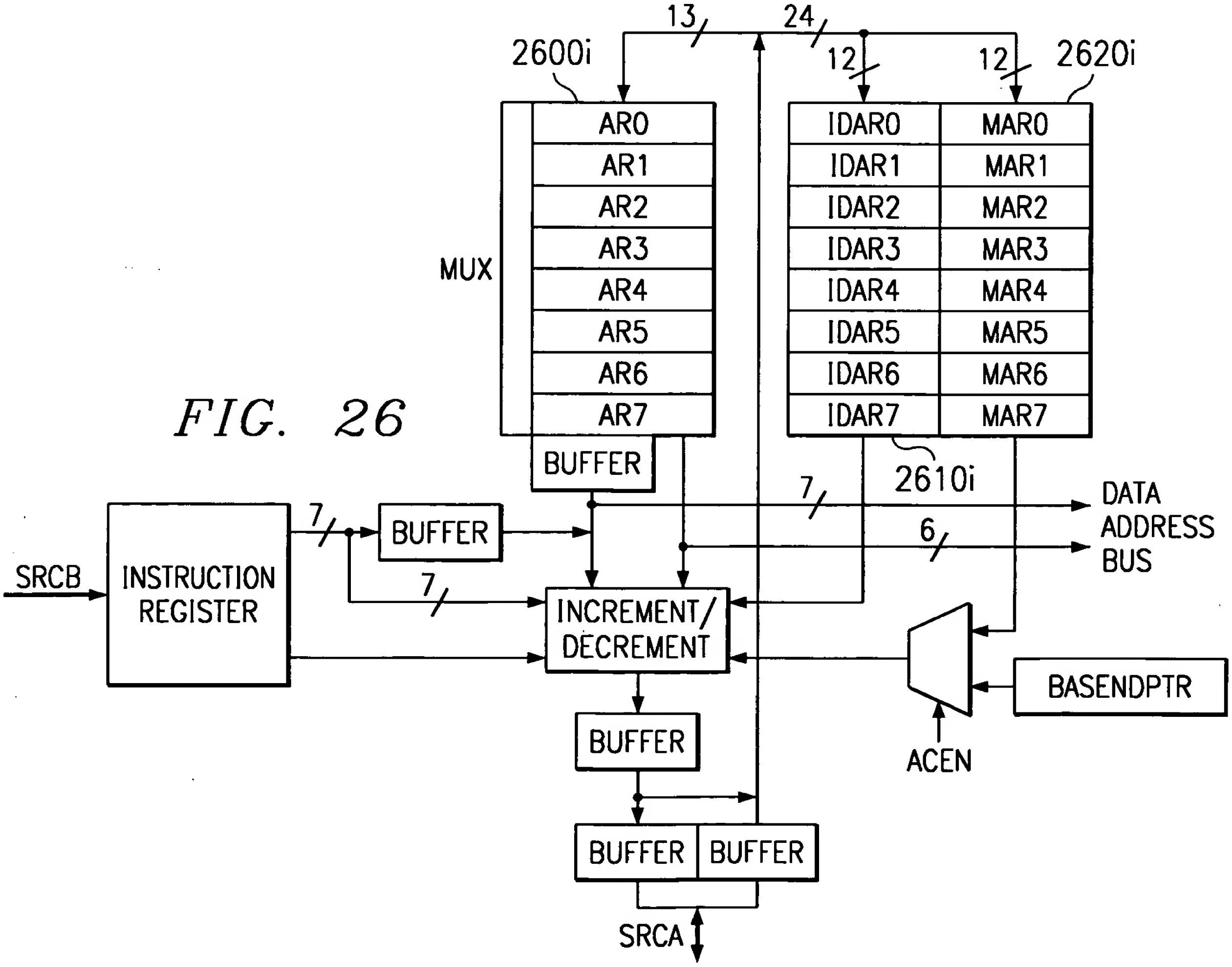

13
24
12
12
2600i
2620i
AR0
AR1
AR2
AR3
AR4
AR5
AR6
AR7
MUX
IDAR0
IDAR1
IDAR2
IDAR3
IDAR4
IDAR5
IDAR6
IDAR7
MAR0
MAR1
MAR2
MAR3
MAR4
MAR5
MAR6
MAR7
2610i
BUFFER
7
6
DATA ADDRESS BUS
SRCB
INSTRUCTION REGISTER
7
BUFFER
7
INCREMENT/ DECREMENT
BASENDPTR
ACEN
BUFFER
BUFFER
BUFFER
SRCA

FIG. 26

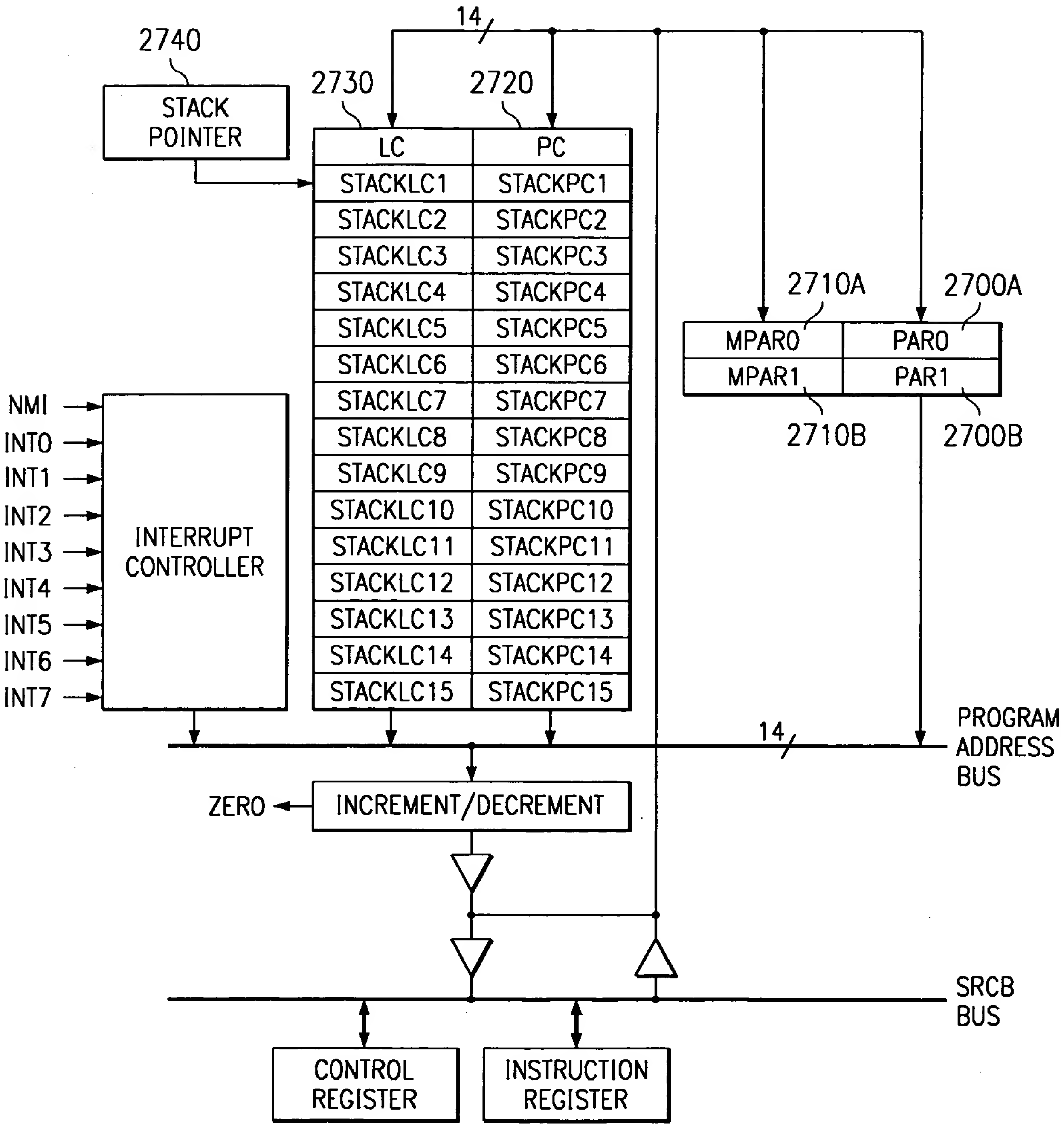

14
2740
STACK POINTER
2730
2720
LC
PC
STACKLC1
STACKPC1
STACKLC2
STACKPC2
STACKLC3
STACKPC3
STACKLC4
STACKPC4
STACKLC5
STACKPC5
STACKLC6
STACKPC6
STACKLC7
STACKPC7
STACKLC8
STACKPC8
STACKLC9
STACKPC9
STACKLC10
STACKPC10
STACKLC11
STACKPC11
STACKLC12
STACKPC12
STACKLC13
STACKPC13
STACKLC14
STACKPC14
STACKLC15
STACKPC15
2710A
2700A
MPAR0
PAR0
MPAR1
PAR1
2710B
2700B
NMI
INT0
INT1
INT2
INT3
INT4
INT5
INT6
INT7
INTERRUPT CONTROLLER
14
PROGRAM ADDRESS BUS
ZERO
INCREMENT/DECREMENT
SRCB BUS
CONTROL REGISTER
INSTRUCTION REGISTER

FIG. 27

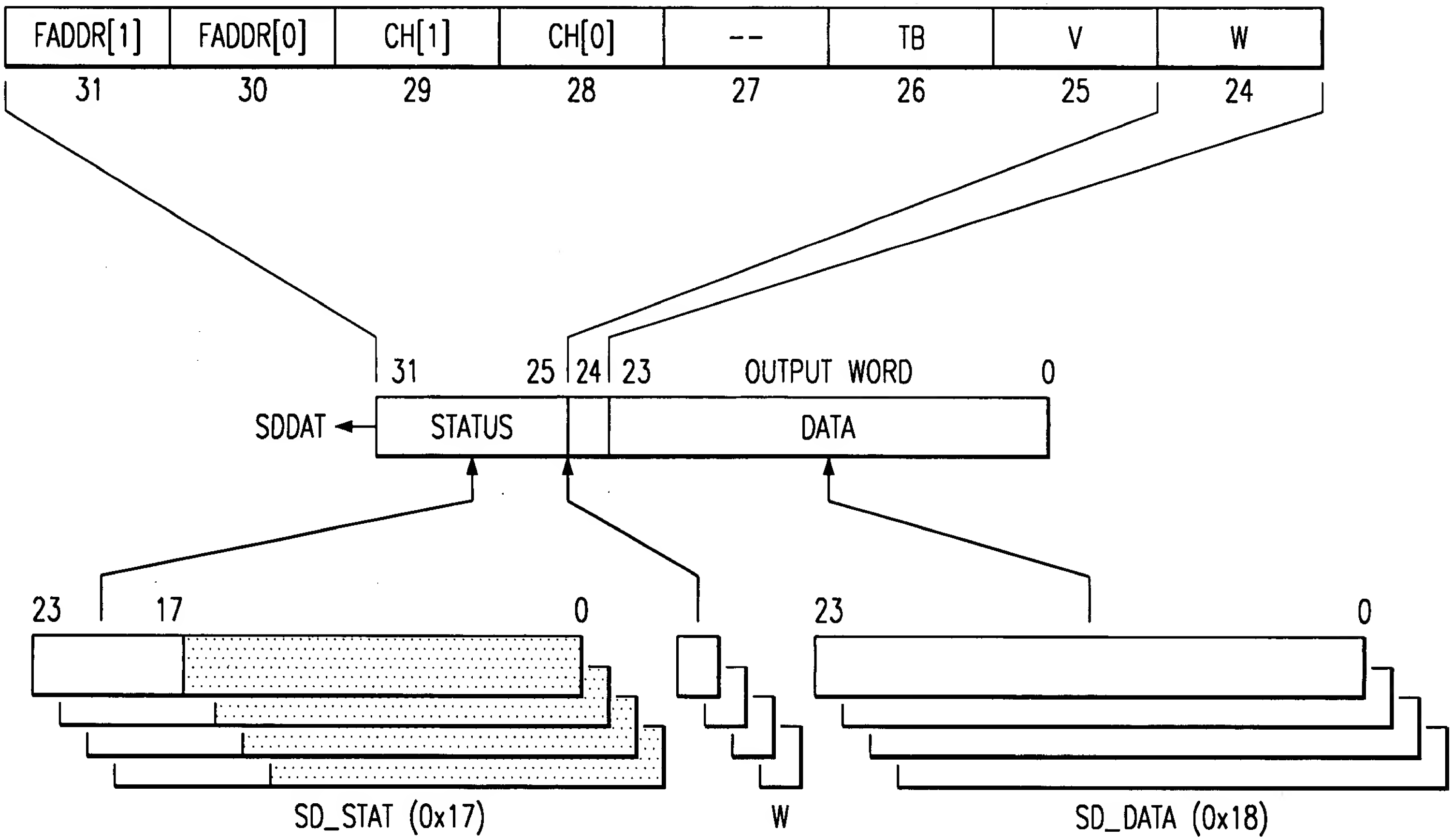

FADDR[1]
FADDR[0]
CH[1]
CH[0]
--
TB
V
W
31
30
29
28
27
26
25
24
31
25
24
23
OUTPUT WORD
0
SDDAT
STATUS
DATA
23
17
0
23
0
SD_STAT (0x17)
W
SD_DATA (0x18)

FIG. 28

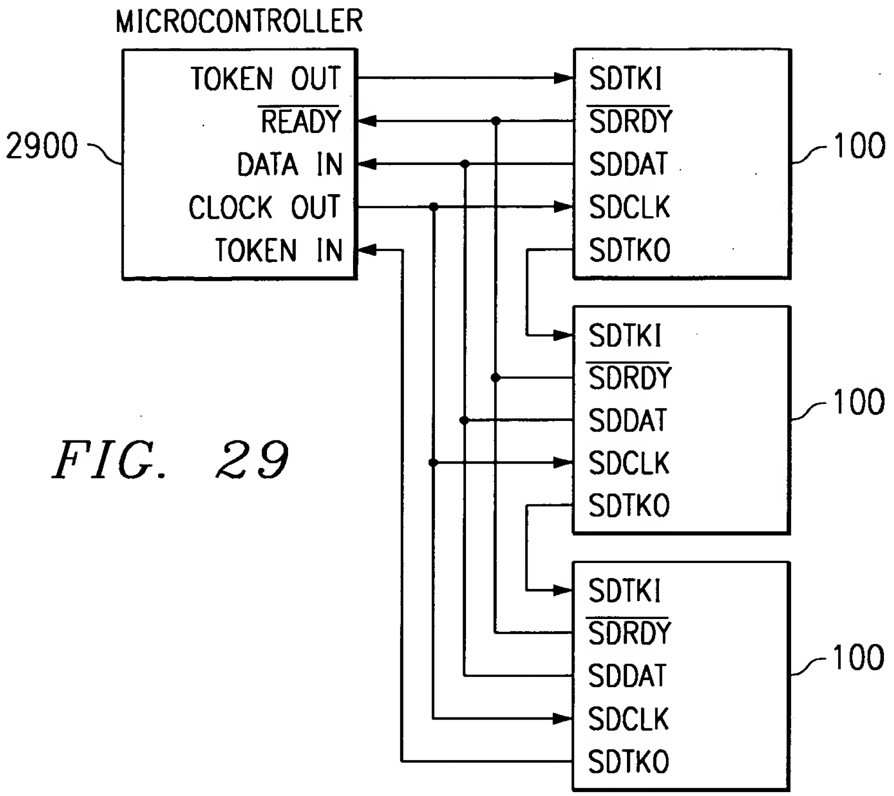


FIG. 29

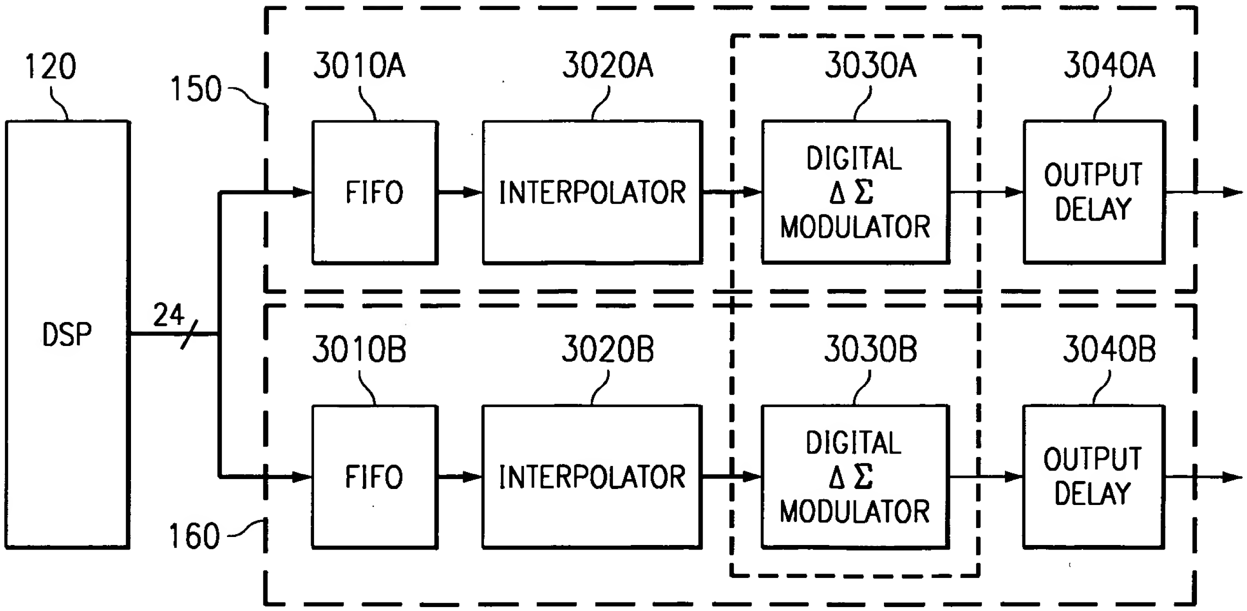


FIG. 30A

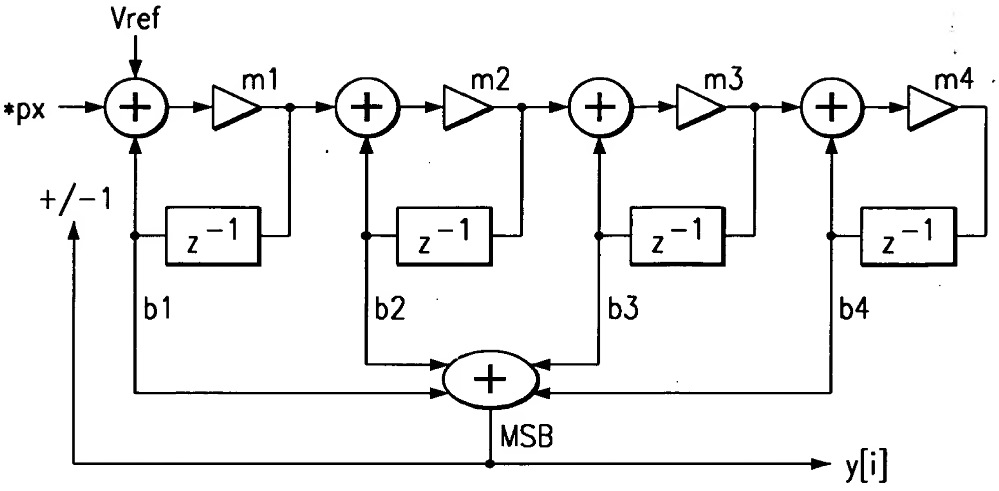


FIG. 30B

FIG. 30C-1

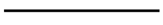


FIG. 30C-2

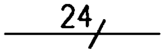


FIG. 30C-3

FIG. 30C-4

MULTIPLEXER

FIG. 30C-5

TRISTATE BUFFER

FIG. 30C-6

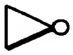


FIG. 30C-7

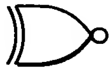


FIG. 30C-8

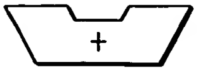


FIG. 30C-9

*
MULTIPLIER

FIG. 30C-10

RIGHT SHIFTER

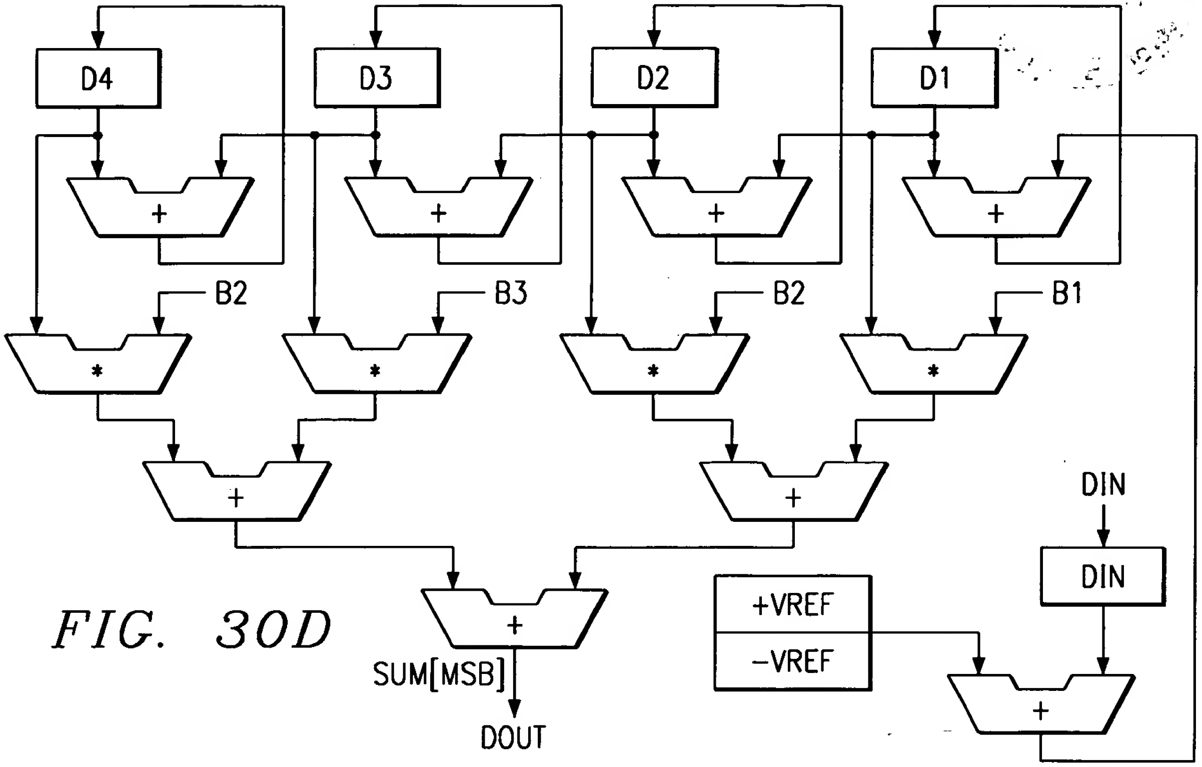


FIG. 30D

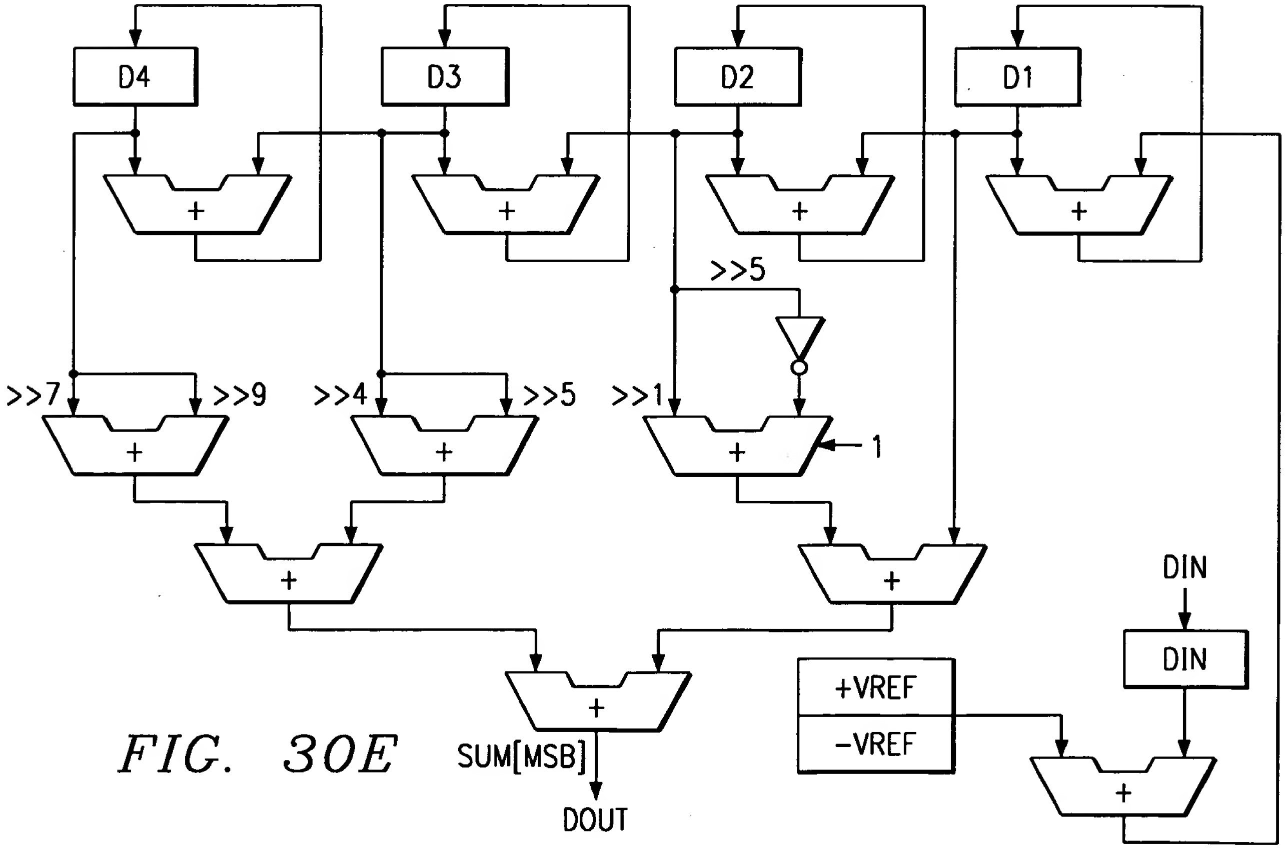


FIG. 30E

FIG. 30F-1

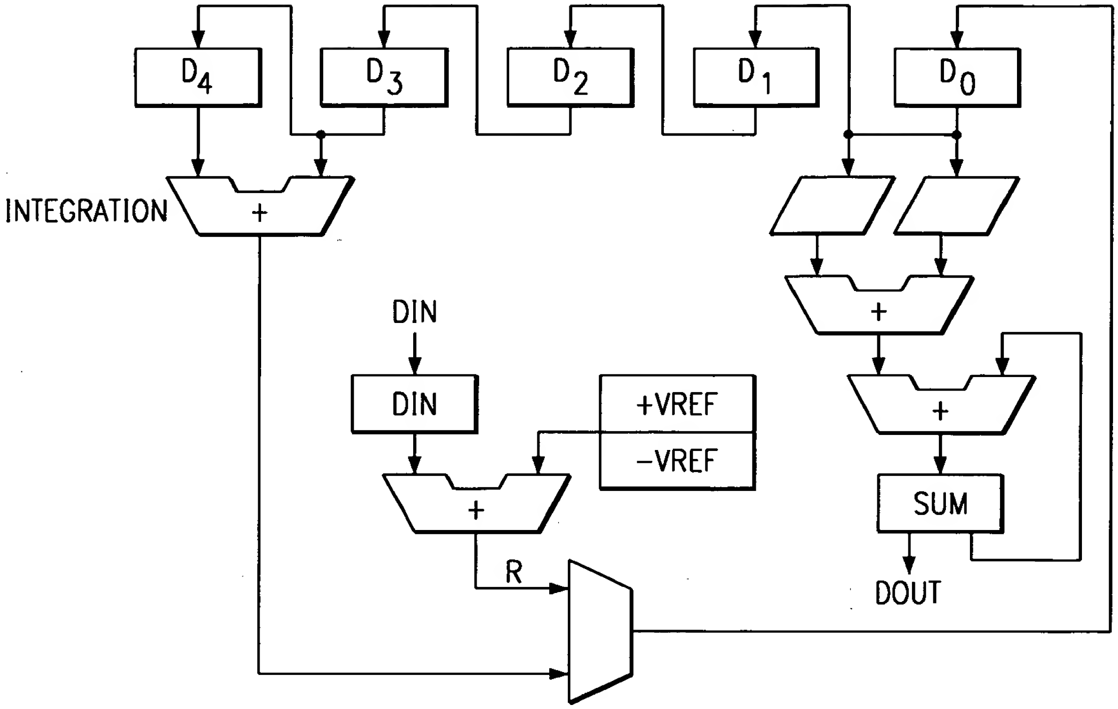


FIG. 30F-2

STATE	ACTIONS DURING STATE		
S0	$D_0(D4_k) = D_4(D4_{k-1})+D_3(D3_{k-1})$	CLEAR SUM	LOAD DIN_k
S1	$D_0(D3_k) = D_4(D3_{k-1})+D_3(D2_{k-1})$	SUM_k += $D_0(D4_k)$>>Shift4	
S2	$D_0(D2_k) = D_4(D2_{k-1})+D_3(D1_{k-1})$	SUM_k += $D_0(D3_k)$>>Shift3	
S3	$D_0(D1_k) = D_4(D1_{k-1})+D_3(R_{k-1})$	SUM_k += $D_0(D2_k)$>>Shift2	
S4		SUM_k += $D_0(D1_k)$>>Shift1	
S5	$D_0(R_k) = DIN_k$ +/− VREF		

FIG. 30G-1

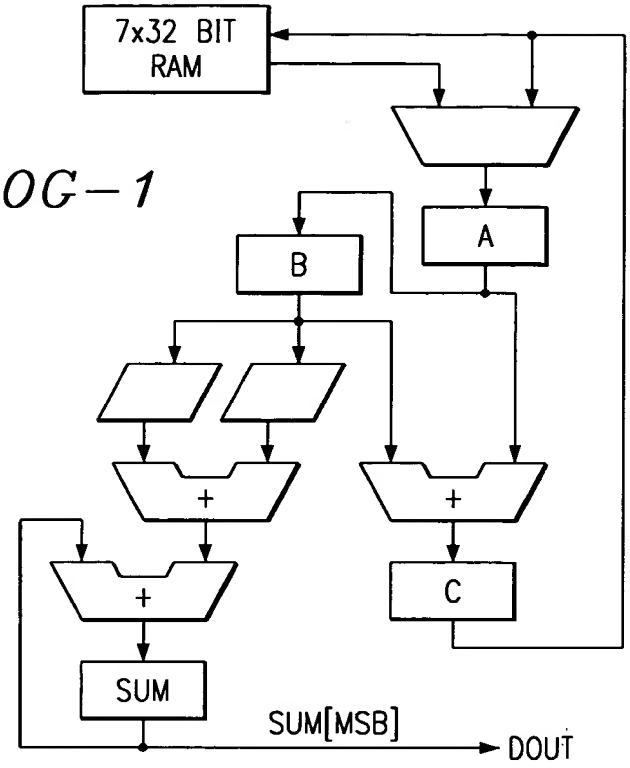
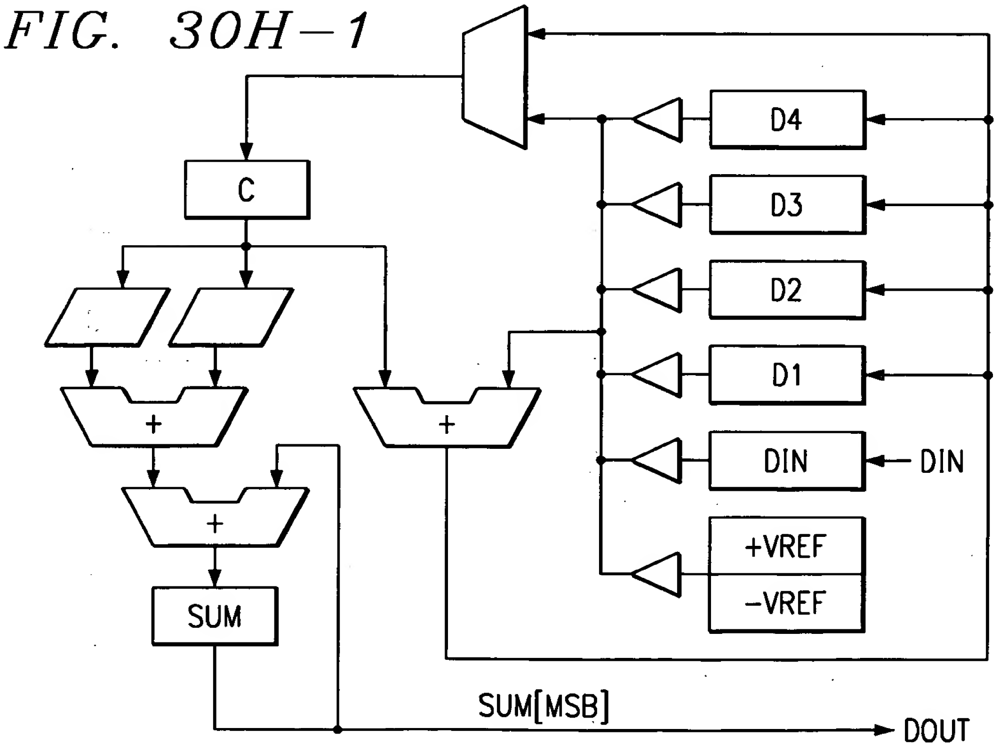


FIG. 30H-1



STATE	ACTIONS DURING STATE			
S0	CLEAR SUM	CLEAR C	CLEAR B	CLEAR A
S1				LOAD A<Mem(D4$_k$)
S2			SHIFT B<A(D4$_k$)	LOAD A<Mem(D3$_k$)
S3	SUM$_k$ += B(D4$_k$)>>Shift4	C = B(D4$_k$) + A(D3$_k$)	SHIFT B<A(D3$_k$)	LOAD A<Mem(D2$_k$)
S4				STORE C>Mem(D4$_{k+1}$)
S5	SUM$_k$ += B(D3$_k$)>>Shift3	C = B(D3$_k$) + A(D2$_k$)	SHIFT B<A(D2$_k$)	LOAD A<Mem(D1$_k$)
S6				STORE C>Mem(D3$_{k+1}$)
S7	SUM$_k$ += B(D2$_k$)>>Shift2	C = B(D2$_k$) + A(D1$_k$)	SHIFT B<A(D1$_k$)	LOAD A<Mem(DIN$_k$)
S8				STORE C>Mem(D2$_{k+1}$)
S9	SUM$_k$ += B(D1$_k$)>>Shift1	C = B(D1$_k$) + A(DIN$_k$)	SHIFT B<A(DIN$_k$)	LOAD A<Mem(VREF)
S10			SHIFT B<A(VREF)	LOADREG A<C(TEMP)
S11		C = +/- B(VREF) + A(TEMP)		
S12				STORE C>Mem(D1$_{k+1}$)

FIG. 30G–2

STATE	ACTIONS DURING STATE			
S0	CLEAR SUM	LOAD C < $D4_k$		LOAD DIN_k
S1	SUM_k += $C(D4_k)$>>Shift4	LOAD C < $D3_k$	$D4_{k+1} = C(D4_k) + D3_k$	
S2	SUM_k += $C(D3_k)$>>Shift3	LOAD C < $D2_k$	$D3_{k+1} = C(D3_k) + D2_k$	
S3	SUM_k += $C(D2_k)$>>Shift2	LOAD C < $D1_k$	$D2_{k+1} = C(D2_k) + D1_k$	
S4	SUM_k += $C(D1_k)$>>Shift1	$C(TEMP) = C(D1_k) + DIN_k$		
S5			$D1_{k+1}$ = C(TEMP) +/− VREF	

FIG. 30H–2

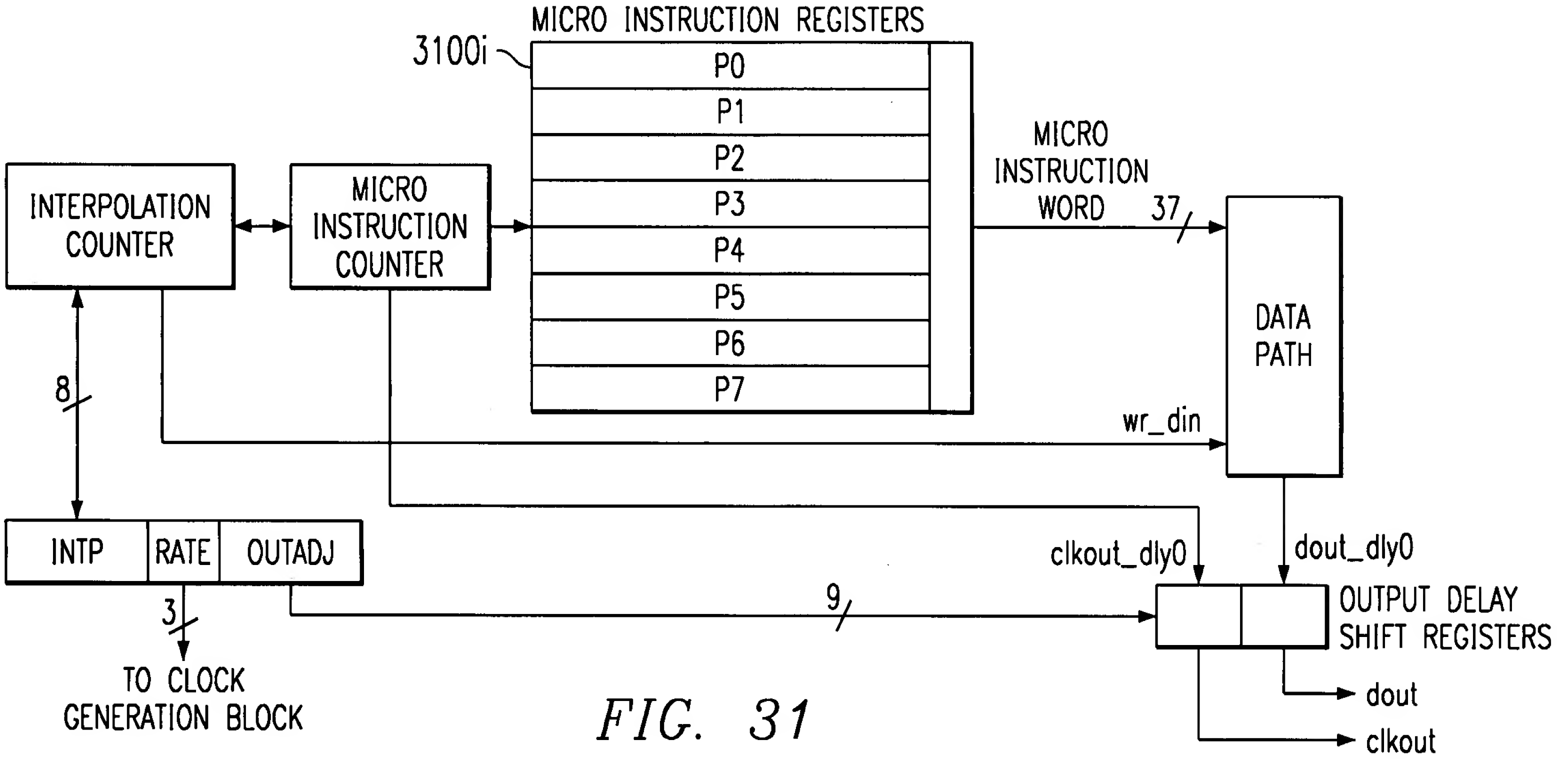

MICRO INSTRUCTION REGISTERS
3100i
P0
P1
P2
P3
P4
P5
P6
P7
INTERPOLATION COUNTER
MICRO INSTRUCTION COUNTER
MICRO INSTRUCTION WORD
37
DATA PATH
wr_din
8
INTP
RATE
OUTADJ
3
TO CLOCK GENERATION BLOCK
9
clkout_dly0
dout_dly0
OUTPUT DELAY SHIFT REGISTERS
dout
clkout

FIG. 31

SCANNED, #17

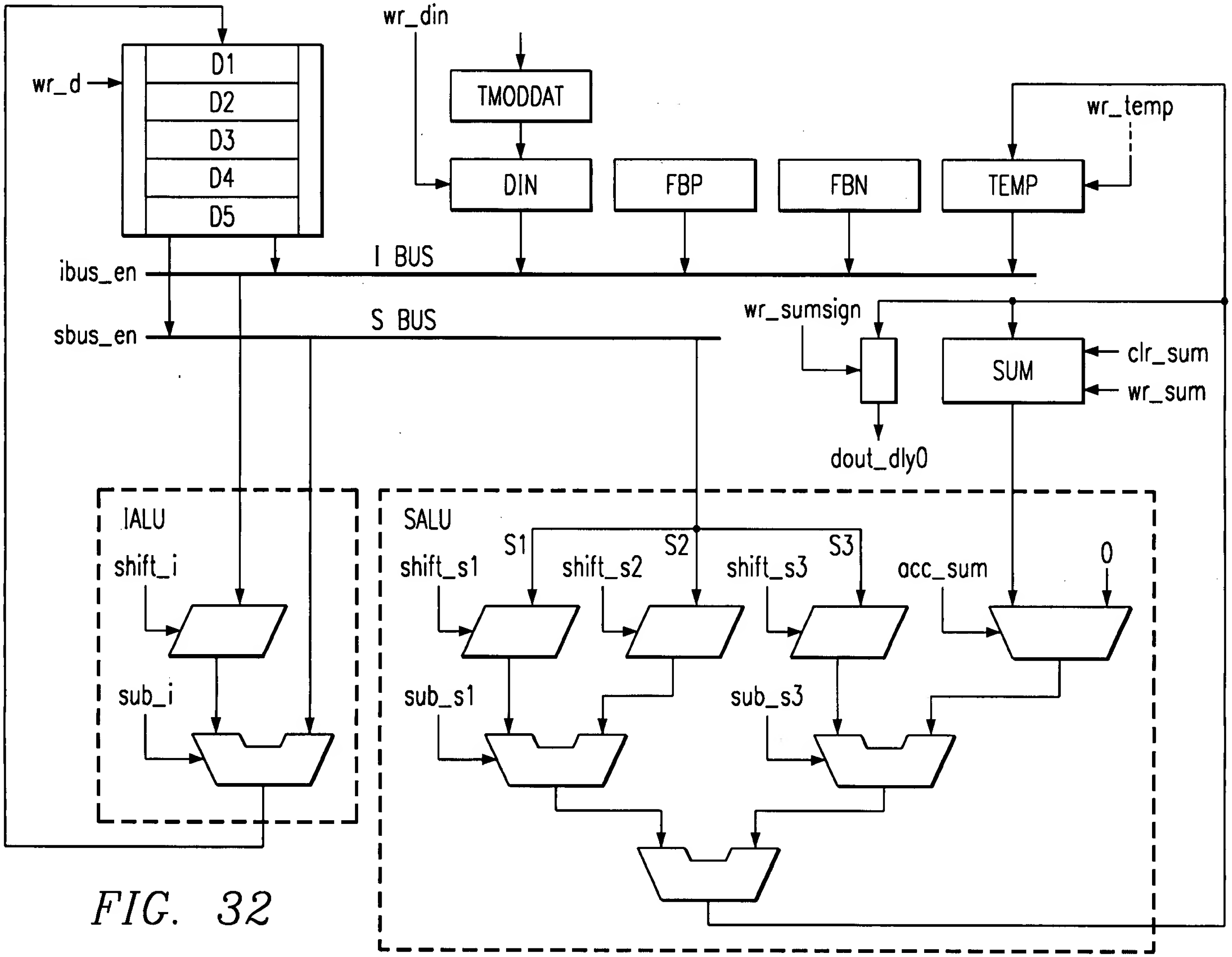


FIG. 32

P	Feedforward	INTEGRATION	TEMP	DIN	SUM	SUMSIGN	TEMP	S BUS	I BUS	WRITE I
0	$SUM_k = D4_k>>11$ $+ D4_k>>9$ $+ D4_k>>7$	$D4_{k+1} = D4_k + D3_k$		LOAD DIN_k	WRITE			+D4>>7 +D4>>9 +D4>>11	+D3	D4
1	$SUM_k = SUM_k$ $+ D3_k>>8$ $+ D3_k>>5$ $+ D3_k>>4$	$D3_{k+1} = D3_k + D2_k$			ACC./ WRITE			+D3>>4 +D3>>5 +D3>>8	+D2	D3
2	$SUM_k = SUM_k$ $+ D2_k>>1$ $= D2_k>>7$ $= D2_k>>4$	$D2_{k+1} = D2_k + D1_k$			ACC./ WRITE			−D2>>4 +D2>>1 −D2>>7	+D1	D2
3	$SUM_k = SUM_k$ $+ D1_k$	$D1_{k+1'} = D1_k + DIN_k$			ACC./ WRITE	WRITE		+D1 +D1 −D1	+DIN	D1
4		$D1_{k+1} = D1_{k+1'}$ +/− VREF							+FB	D1
5										
6										
7										

FIG. 33

FIG. 34A

G	C	S	T	wr_d			fbus_en			sbus_en			shift_i					s i	shift_s1					s s 1	shift_s2					shift_s3					s s 3	A'	HEX
36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0	
0	0	1	0	1	0	0	0	1	1	1	0	0	0	0	0	0	0	0	0	0	1	1	1	0	0	1	0	0	1	0	1	0	1	1	0	0	051C 00E4AC
0	0	1	0	0	1	1	0	1	0	0	1	1	0	0	0	0	0	0	0	0	1	0	0	0	0	0	1	0	1	0	1	0	0	0	0	1	04D3 0082A1
0	0	1	0	0	1	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1	0	0	1	0	0	0	0	1	0	0	1	1	1	1	1	048A 00909F
1	0	1	0	0	0	1	1	0	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1469 000003
0	0	0	0	0	0	1	1	1	1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0079 000000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 000000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 000000
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0000 000000

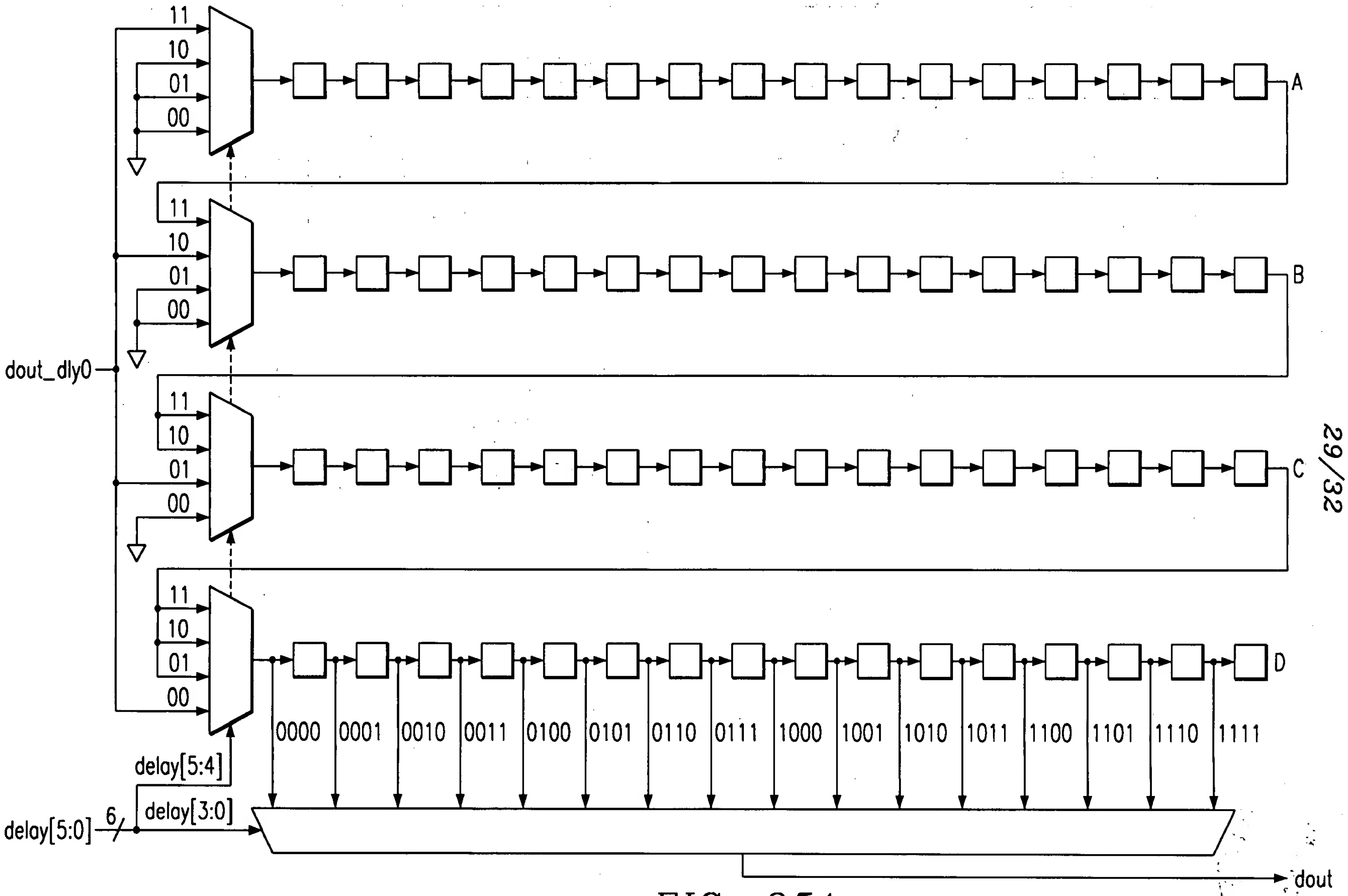

11
10
01
00
A
B
C
D
dout_dly0
delay[5:4]
delay[3:0]
delay[5:0]
6
0000
0001
0010
0011
0100
0101
0110
0111
1000
1001
1010
1011
1100
1101
1110
1111
dout

FIG. 35A

dout_dly0	DATA OUTPUT BIT, 0 DELAY
dout	DATA OUTPUT BIT, 0-63 CLOCK DELAY
delay[5:0]	HOW MANY CLOCKS (0-63) TO DELAY OUTPUT DATA dout_dly0
delay[5:4]	SELECTS SEGMENT INTO WHICH TO DIRECT dout_dly0
delay[3:0]	SELECTS WHERE TO TAP SEGMENT D TO GET dout

FIG. 35B

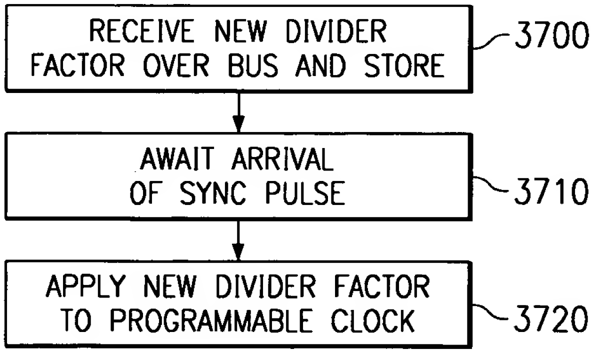

RECEIVE NEW DIVIDER FACTOR OVER BUS AND STORE
3700
AWAIT ARRIVAL OF SYNC PULSE
3710
APPLY NEW DIVIDER FACTOR TO PROGRAMMABLE CLOCK
3720

FIG. 37

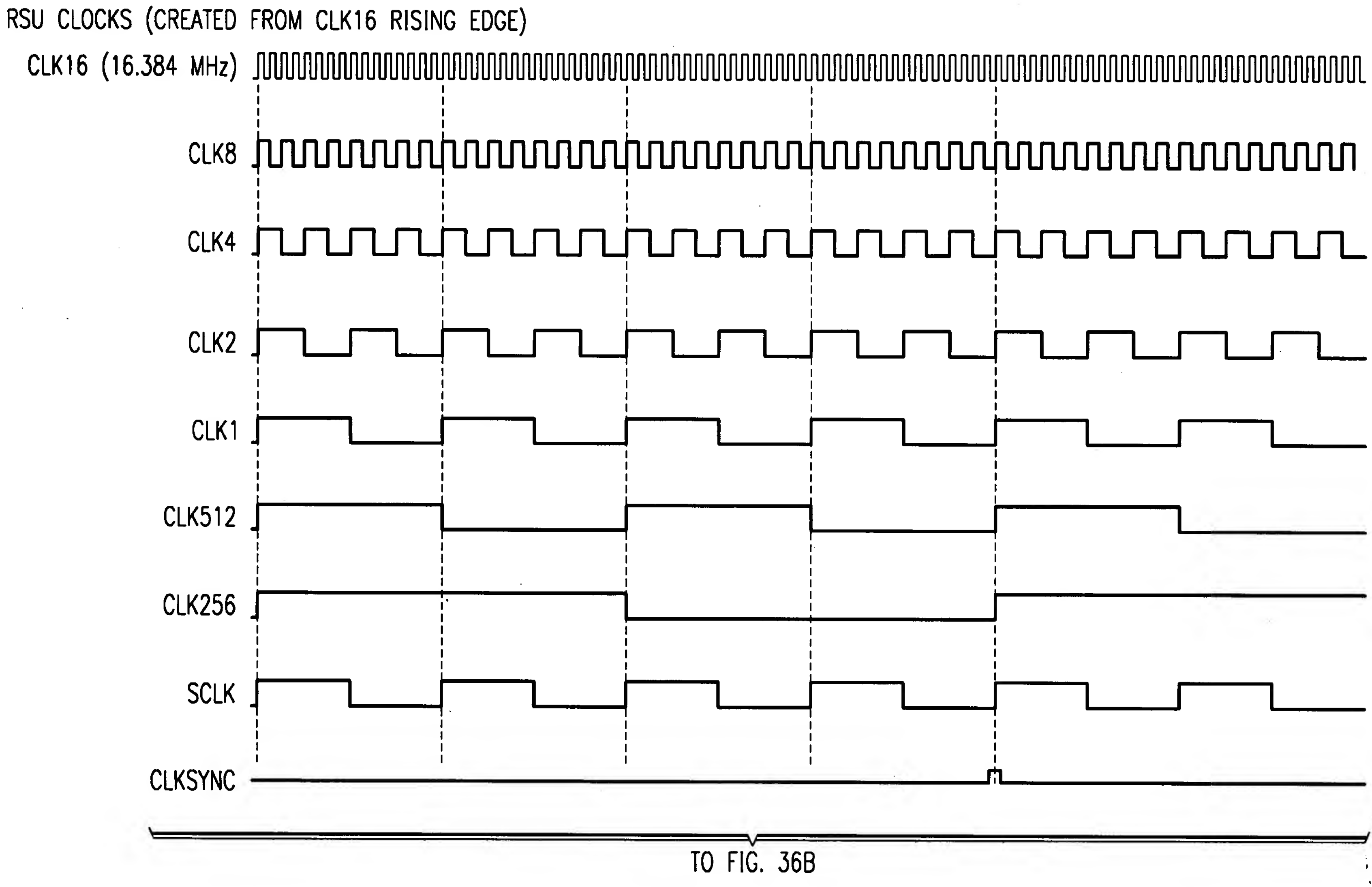

RSU CLOCKS (CREATED FROM CLK16 RISING EDGE)
CLK16 (16.384 MHz)
CLK8
CLK4
CLK2
CLK1
CLK512
CLK256
SCLK
CLKSYNC
TO FIG. 36B

FIG. 36A

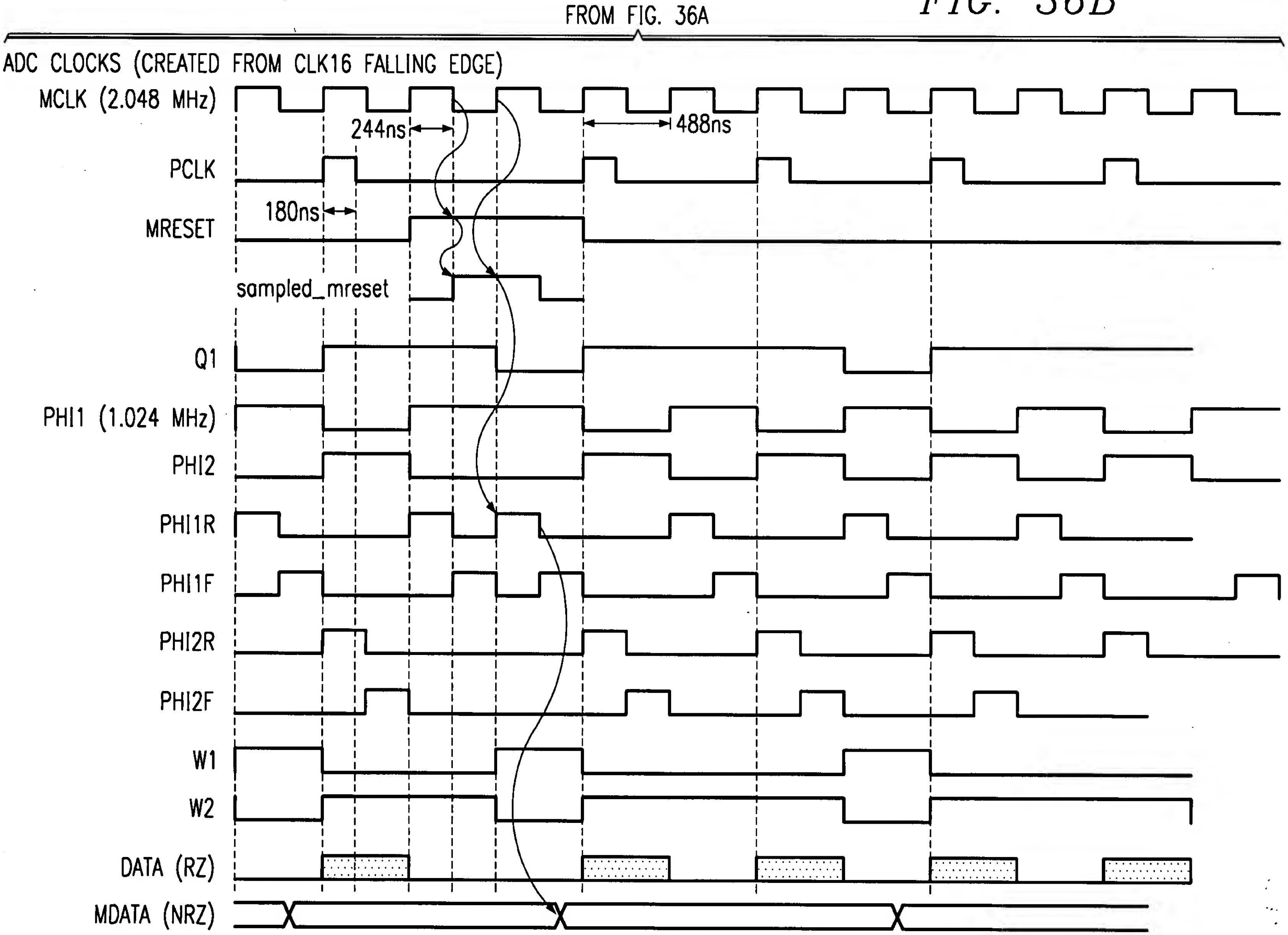

FROM FIG. 36A
ADC CLOCKS (CREATED FROM CLK16 FALLING EDGE)
MCLK (2.048 MHz)
244ns
488ns
PCLK
180ns
MRESET
sampled_mreset
Q1
PHI1 (1.024 MHz)
PHI2
PHI1R
PHI1F
PHI2R
PHI2F
W1
W2
DATA (RZ)
MDATA (NRZ)

FIG. 36B